ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗΝ ΒΙΟΙΑΤΡΙΚΗ

ΠΡΟΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΣΠΟΥΔΩΝ

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Διατήρηση ιδιωτικότητας δεδομένων κίνησης κατά την ομαδοποίηση συναγερμών συστημάτων ανίχνευσης εισβολών σε δίκτυα πολλαπλών οργανισμών

Γεώργιος Κ. Θεοδωρίδης

ΛΑΜΙΑ

ΦΕΒΡΟΥΑΡΙΟΣ 2018

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Διατήρηση ιδιωτικότητας δεδομένων κίνησης κατά την ομαδοποίηση συναγερμών συστημάτων ανίχνευσης εισβολών σε δίκτυα πολλαπλών οργανισμών

Γεώργιος Κ. Θεοδωρίδης

ΕΠΙΒΛΕΠΩΝ ΚΑΘΗΓΗΤΗΣ
Γεώργιος Σπαθούλας
Μέλος ΕΔΙΠ
Πανεπιστήμιο Θεσσαλίας

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ ΠΑΡΑΚΟΛΟΥΘΗΣΗΣ

Γεώργιος Σπαθούλας
Μέλος ΕΔΙΠ
Πανεπιστήμιο Θεσσαλίας

Μαρία Αδάμ
Αναπληρώτρια Καθηγήτρια
Πανεπιστήμιο Θεσσαλίας

Ιωάννης Αναγνωστόπουλος
Αναπληρωτής Καθηγητής
Πανεπιστήμιο Θεσσαλίας

Ημερομηνία Εξέτασης: 7 Φεβρουαρίου 2018

ΠΕΡΙΛΗΨΗ

Σήμερα πολλοί οι οργανισμοί έρχονται αντιμέτωποι με πολλαπλές και σοβαρές επιθέσεις ασφαλείας. Για αυτό τον λόγο πολλοί από αυτούς τους οργανισμούς επενδύουν χρήματα και καταβάλλουν προσπάθειες στην εγκατάσταση συστημάτων ανίχνευσης εισβολών, προκειμένου να παρακολουθούν τέτοιες επιθέσεις. Λόγω της παγκόσμιας φύσης τέτοιων επιθέσεων, θα ήταν επωφελής για τέτοιες οργανώσεις κάποιου είδους συνεργασία, ώστε να αξιολογούν καλύτερα τη σοβαρότητα και τη σημασία κάθε επίθεσης. Από την άλλη ”πλευρά”, είναι αδύνατο για αυτούς να ανταλλάσσουν δεδομένα, όπως η κυκλοφορία του δικτύου τους ή οι δικές τους ειδοποιήσεις ανίχνευσης εισβολών λόγω θεμάτων ιδιωτικότητας. Σε αυτή την εργασία προτείνεται ένα πρωτόκολλο συνεργασίας για την ανίχνευση επιθέσεων αλλά και παράλληλα τη διατήρηση της ιδιωτικότητας των οργανισμών. Συγκεκριμένα, χρησιμοποιείται ομομορφική κρυπτογράφηση για την ομαδοποίηση των συναγερμών σε ένα δίκτυο οργανισμών με την χρήση μίας έμπιστης τρίτης οντότητας.

ΘΕΜΑΤΙΚΗ ΠΕΡΙΟΧΗ: Ασφάλεια Δικτύων,Κρυπτογραφία και Ιδιωτικότητα

ΛΕΞΕΙΣ ΚΛΕΙΔΙΑ: Ανίχνευση εισβολών, ομαδοποίηση, ομομορφική κρυπτογράφιση, ιδιωτικότητα

ABSTRACT

Nowadays a lot of organizations are coping with multiple and severe security attacks and they invest money and effort into installing intrusion detection systems in order to monitor such attacks. Due to the global nature of such attacks it would be beneficial for such organizations to cooperate in order to better assess the severity and the importance of each attack. On the other hand it is impossible for them to exchange data such as their network traffic or their intrusion detection alerts due to privacy reasons. A privacy preserving cooperation protocol for attacks detection is proposed in this paper. Specifically homomorphic encryption is used to perform alerts clustering at an inter-organizational level with the use of an honest but curious trusted third party.

SUBJECT AREA: Security, Cryptography and Privacy

KEYWORDS: Intrusion detection, clustering, homomorphic encryption, privacy

Στους αγαπημένους μου.

ΕΥΧΑΡΙΣΤΙΕΣ

Πριν προχωρήσω παρακάτω, θέλω πραγματικά να ευχαριστήσω ορισμένους από τους ανθρώπους που γνώρισα και συνεργάστηκα μαζί τους για την πραγματοποίηση της πτυχιακής μου εργασίας αλλά και τους αφανείς ήρωες που είχα στο πλευρό μου κατά την διάρκεια των φοιτητικών μου χρόνων.

Πρώτα από όλους θέλω να ευχαριστήσω τον επιβλέποντα της πτυχιακής μου εργασίας, διδάσκοντα Γεώργιο Σπαθούλα για την πίστη που έδειξε στο πρόσωπο μου και την συνεχή καθοδήγηση που μου παρείχε σε όλη την διάρκεια της έρευνας και εκπόνησης της εργασίας.

Στη συνέχεια θα ήθελα να ευχαριστήσω έναν άνθρωπο ο οποίος συνέβαλε καθοριστικά στην ολοκλήρωση αυτής της εργασίας. Δεν είναι άλλος από τον συμφοιτητή μου Γεώργιο Δαμίρη. Ο Γιώργος ήταν πάντα εκεί για να ισορροπεί τον ενθουσιώδη χαρακτήρα μου με την πραότητα του και δεν νομίζω πως θα μπορούσα να έχω κάποιον καλύτερο συνεργάτη.

Δεν θα μπορούσα να παραλείψω να ευχαριστήσω τους καθηγητές της σχολής που με υπομονή τόσα χρόνια συνέβαλαν στην απόκτηση των απαραίτητων γνώσεων για την επιτυχή φοίτησή μου και την εκπόνηση της πτυχιακής μου εργασίας, αλλά κυρίως που ενίσχυσαν την αγάπη μου για την επιστήμη και την έρευνα, που τώρα πια έχει γίνει αναπόσπαστο κομμάτι της καθημερινότητας μου.

Τέλος, θέλω να ευχαριστήσω την οικογένεια μου και ιδιαίτερα τους γονείς μου και την γιαγιά μου, για την βοήθεια τους όλα αυτά τα χρόνια. Σας ευχαριστώ για την ψυχολογική αλλά και οικονομική υποστήριξη που μου προσφέρατε.

Γ.Θ

ΠΕΡΙΕΧΟΜΕΝΑ

ΛΙΣΤΑ ΣΧΗΜΑΤΩΝ

ΛΙΣΤΑ ΠΙΝΑΚΩΝ

ΠΡΟΛΟΓΟΣ

Η πτυχιακή μου εργασία πάνω στη Διατήρηση Ιδιωτικότητας Δεδομένων Κίνησης κατά την ομαδοποίηση συναγερμών συστημάτων ανίχνευσης εισβολών σε δίκτυα πολλαπλών οργανισμών, υλοποιήθηκε στα πλαίσια των σπουδών μου στο Τμήμα Πληροφορικής με εφαρμογές στην Βιοϊατρική της Σχολής Θετικών Επιστημών του Πανεπιστημίου Θεσσαλίας. Η έρευνα και υλοποίηση του πειράματος πραγματοποιήθηκε κατά την διάρκεια του εαρινού εξαμήνου του 2017. Η συγγραφή της πτυχιακής εργασίας κατά την διάρκεια του χειμερινού εξαμήνου του 2018. Η ιδέα για το συγκεκριμένο έργο εμφανίστηκε κατόπιν συζήτησης με τον καθηγητή του τμήματος και επιβλέπων καθηγητή της πτυχιακής εργασίας, Γεώργιο Σπαθούλα.

Δηλώνω υπεύθυνα ότι είμαι ο συγγραφέας αυτής της πτυχιακής εργασίας και ότι κάθε βοήθεια την οποία είχα για την προετοιμασία της, είναι πλήρως αναγνωρισμένη και αναφέρεται στην εργασία.

Επίσης, έχω κάνει σαφής αναφορές τις όποιες πηγές από τις οποίες έκανα χρήση δεδομένων, προτάσεων, ιδεών ή λέξεων, είτε αυτές αναφέρονται ακριβώς είτε είναι παραφρασμένες.

Λαμία, 2018-01-04

Γεώργιος Θεοδωρίδης

1. ΕΙΣΑΓΩΓΗ

Οι νέες τεχνολογίες σχετικά με το διαδίκτυο κάνουν την ζωή μας πιο εύκολη και τις διεργασίες πιο αποτελεσματικές τόσο στον ιδιωτικό όσο και στον επιχειρηματικό τομέα. Ωστόσο, αυτή η παγκόσμια διασύνδεση έχει ένα μεγάλο μειονέκτημα, καθώς ”οτιδήποτε” συνδεδεμένο στο διαδίκτυο αποτελεί αυτόματα έναν πιθανό στόχο κυβερνοεπιθέσεων ή θα μπορούσε να χρησιμοποιηθεί για να συμμετάσχει σε κάποια εγκληματική ενέργεια του κυβερνοχώρου.

Ως εκ τούτου, η ασφάλεια δικτύου κάθε μέρα γίνεται όλο και πιο αναγκαία. Αυτό ενισχύεται ιδιαίτερα από τις νέες τεχνολογικές τάσεις όπως το Internet of Things και τη μεγάλη γκάμα συστημάτων και υπηρεσιών βασισμένων στο Cloud. Οι παραπάνω τεχνολογίες έχουν ως αποτέλεσμα την μετάβαση του διαδικτύου από το συμβατικό σύστημα Client-Server σε μία διαφορετική δομή όπου συστήματα έξυπνων ή/και κινητών συσκευών άμεσα συνδεδεμένων στο διαδίκτυο, ορίζουν περισσότερο πολύπλοκες δομές.

Δυστυχώς, η συντήρηση και η διαφύλαξη της ασφάλειας παραμελούνται συχνά στη χρήση αυτών των συσκευών, καθιστώντας τις επιρρεπείς σε ανεπιθύμητες διεισδύσεις από κακόβουλους εισβολείς. Παράλληλα ο αριθμός των ευάλωτων συσκευών αυξάνεται συνεχώς, λόγω της αυξανόμενης αποδοχής που απολαμβάνουν από τους τελικούς χρήστες. Είναι όλο και πιο σύνηθες να πρέπει να διαχειριστούμε ετερογενείς δικτυακές υποδομές στις οποίες συμμετέχουν αυτές οι συσκευές (π.χ. ασύρματες εστίες όπως εστιατόρια, πανεπιστήμια, χώροι εργασίας ή ακόμη και αυτοκίνητα). Ως εκ τούτου, οι δυναμικές αλλαγές τόσο στο μέγεθος όσο και στην δομή των υπό προστασία δικτύων γίνονται ο κανόνας και όχι η εξαίρεση. Σε αυτές τις περιπτώσεις είναι πολύ πιθανό να διεισδύσει μια συσκευή της οποίας κάποιος κακόβουλος επιτιθέμενος έχει τον έλεγχο και έτσι να καταστεί η ίδια η δικτυακή υποδομή ευάλωτη σε περαιτέρω επιθέσεις.

Για παράδειγμα, τον Οκτώβριο του 2016 ξεκίνησε μια συντονισμένη επίθεση distributed-denial-of-service (DDoS) ενάντια σε έναν μεγάλο DNS πάροχο, κάνοντας ανέφικτη την επικοινωνία με έναν σημαντικό αριθμό υπηρεσιών διαδικτύου στη Βόρεια Αμερική και σε άλλα μέρη. Η πλειοψηφία των επιτιθέμενων συσκευών δεν ήταν συμβατικοί υπολογιστές, αλλά ανεπαρκώς προστατευμένες και ευάλωτες συσκευές, συνδεδεμένες στο διαδίκτυο, όπως κάμερες IP-CCTV, routers και έξυπνες τηλεοράσεις. Επιπλέον, τον Νοέμβριο του 2016, η Deutsche Telekom έκανε αναφορά για ύποπτη συμπεριφορά στις υποδομές της, διακόπτοντας κάποιες διαδικτυακές υπηρεσίες της. Διαπιστώθηκε ότι πάνω από 900.000

συσκευές δρομολόγησης των πελατών είχαν μολυνθεί από κακόβουλο λογισμικό εκτελέσιμων αρχείων. Επειδή και οι δύο παραβιάσεις ασφαλείας είχαν σχετικές ομοιότητες, δεν μπορεί να αποκλειστεί ότι και οι δύο περιπτώσεις σχετίζονται με την ίδια ή παρόμοιες παγκόσμιες εκστρατείες μόλυνσης IoT συσκευών.

Με αυτές τις εξελίξεις, η ανάγκη για συστήματα ανίχνευσης εισβολών (IDS) και συνεργατικά IDS (CIDS) είναι πια μεγάλη. Η συγκεντρωτική μελέτη και ανάλυση των περιστατικών ασφαλείας περισσοτέρων του ενός οργανισμών θα μπορούσε να οδηγήσει σε σωστότερα και συντομότερα συμπεράσματα σχετικά με τις πιθανές επιθέσεις. Παράλληλα όμως θα έθετε σε κίνδυνο την ιδιωτικότητα των δικτυακών δεδομένων των οργανισμών.

Στο πλαίσιο της παρούσας πτυχιακής, μελετήθηκε η δυνατότητα εκτέλεσης ενός αλγορίθμου ομαδοποίησης συναγερμών συστημάτων ανίχνευσης δικτυακών εισβολών διαφορετικών οργανισμών (clients). Βασική προϋπόθεση λειτουργίας του αλγορίθμου αποτελεί η ελάχιστη δυνατή απώλεια ιδιωτικότητας όσον αφορά τα δεδομένα κίνησης των οργανισμών. Για να επιτευχθεί αυτό, σχεδιάστηκε ένα συνεργατικό σύστημα ομαδοποίησης συναγερμών εισβολών με την χρήση μιας τρίτης έμπιστης οντότητας. Το σύστημα επιτυγχάνει την εξαγωγή χρήσιμων συμπερασμάτων και την μείωση της πιθανότητας λάθος συναγερμών, χωρίς όμως να προσβάλλεται σημαντικά η ιδιωτικότητα των δεδομένων κίνησης των διαφορετικών συναγερμών.

Σχεδιάστηκε και υλοποιήθηκε το σύστημα PP-CIDS (Privacy Preserving - Collaborative Intrusion Detection System) το οποίο βασίζεται σε μία κατανεμημένη αρχιτεκτονική Client-Server, όπου οι Clients (Οργανισμοί) μοιράζονται τις πληροφορίες (κρυπτογραφημένες) που συλλέγουν από τα δικά τους IDS με έναν Server (έμπιστη τρίτη οντότητα). Αυτός εκτελεί την ομαδοποίηση των συναγερμών και επιστρέφει τα αποτελέσματα στους οργανισμούς, ώστε αυτοί να είναι σε θέση να βγάλουν πιο χρήσιμα συμπεράσματα. Η όλη διαδικασία σχεδιάσθηκε έτσι ώστε ο Server να μην είναι σε θέση να διαβάσει τα δεδομένα των Clients, καθώς και οι Clients να μαθαίνουν την ελάχιστη δυνατή πληροφορία για τα συμβάντα των άλλων μελών του συστήματος.

Χρησιμοποιήθηκαν τεχνικές ομομορφικής κρυπτογράφησης (Paillier cryptosystem) για τους συναγερμούς, ώστε να είναι δυνατή η επεξεργασία τους από μία έμπιστη τρίτη οντότητα (Server), χωρίς να απαιτείται η αποκρυπτογράφησή τους.

Ο αλγόριθμος αποτελείται από τα εξής μέρη:

- Αμφίδρομη επικοινωνία μεταξύ Client και Server (πολυνηματισμός)

- Κρυπτογράφηση συναγερμών από τους Clients
- Αποστολή κρυπτογραφημένων συναγερμών στον Server
- Υπολογισμός αποστάσεων συναγερμών χρησιμοποιώντας την ομομορφική ιδιότητα του Paillier για πράξεις με κρυπτογραφημένα δεδομένα
- Αξιοποίηση της αμφίδρομης επικοινωνίας για την χρήση βοηθητικών συναρτήσεων σύγκρισης και πολλαπλασιασμού των Clients από το Server

ΥΛΟΠΟΙΗΣΗ

Η υλοποίηση του αλγορίθμου έγινε με χρήση της γλώσσας προγραμματισμού Python. Για την υλοποίηση χρησιμοποιήθηκαν οι βιβλιοθήκες Threading και Pyro4 για την υλοποίηση της αμφίδρομης επικοινωνίας, NumPy για υπολογισμούς και πράξεις μεταξύ πινάκων και Paillier για την κρυπτογράφηση των συναγερμών. Επιπλέων υλοποιήθηκε αλγόριθμος δυναμικής σύνθεσης πίνακα από υποπίνακες με την χρήση των μεθόδων hstack, vstack της NumPy. Υλοποιήθηκε αλγόριθμος υπολογισμού των αποστάσεων μεταξύ των συναγερμών. Οι αποστάσεις αυτές χρησιμοποιήθηκαν για την ομαδοποίηση των συναγερμών μεσώ του αλγορίθμου K-medoids.

ΣΥΜΠΕΡΑΣΜΑΤΑ

Η υλοποίηση και η εκτέλεση του αλγορίθμου μας οδήγησαν στα εξής συμπεράσματα:

- Η ιδιωτικότητα των δεδομένων δεν παραβιάζεται κατά την διαδικασία της ομαδοποίησης
- Η έμπιστη τρίτη οντότητα (Server) είναι ικανή να πραγματοποιεί ομαδοποίηση εκτελώντας το μεγαλύτερο ποσοστό των απαραίτητων πράξεων και μετασχηματισμών
- Οι επιμέρους οργανισμοί ενημερώνονται για την συνολική κατάσταση ασφαλείας ανάμεσα σε όλα τα συνεργαζόμενα μέρη
- Μπορούν να σχεδιάσουν πολιτικές ασφαλείας με τα καλύτερα δυνατά αποτελέσματα

ΜΕΛΛΟΝΤΙΚΗ ΕΡΕΥΝΑ

- Χρήση παράλληλων δομών hardware στην πλευρά της έμπιστης τρίτης οντότητας για την βελτίωση του μέγιστου δυνατού ρυθμού επεξεργασίας συναγερμών.
- Ανίχνευση μη έντιμων συμπεριφορών των επιμέρους μελών, μέσω της χρήσης secret sharing schemes.
- Ανάπτυξη μηχανισμού consensus μεταξύ των clients, όσον αφορά τους κανόνες ανίχνευσης εισβολών.
- Χρήση κρυπτογραφικών μεθόδων με σκοπό την διαφύλαξη της ιδιωτικότητας των δεδομένων κίνησης των επιμέρους οργανισμών κατά την παραπάνω διαδικασία.

2. ΣΧΕΤΙΚΗ ΕΡΕΥΝΑ

Από τους πρώτους που μελέτησαν τρόπους συνεργασίας των συστημάτων ανίχνευσης εισβολών ήταν οι Chenfeng Vincent Zhou, Christopher Leckie και Shanika Karunasekera [25]. Στην μελέτη τους πάνω σε ένα αποκεντρωμένο σύστημα διαμοιρασμού συναγερμών IDS, προτείνουν έναν πολυδιάστατο αλγόριθμο ομαδοποίησης συναγερμών για εξαγωγή των σημαντικών προτύπων από τους συναγερμούς. Αρχικά, χρησιμοποιούν έναν αλγόριθμο συσχέτισης δύο φάσεων, ο οποίος πρώτα ομαδοποιεί τους συναγερμούς τοπικά σε κάθε IDS, κι έπειτα αναφέρει σημαντικά πρότυπα συναγερμών σε ένα κοινό δίκτυο συσχέτισης μεταξύ των IDS. Μέσω μιας πιθανοτικής προσέγγισης αποφασίζουν πότε ένα πρότυπο σε τοπικό επίπεδο είναι αρκετά σημαντικό ώστε να συνεχίσει την συσχέτιση σε επίπεδο δικτύου. Στη συνέχεια εφαρμόζουν τον παραπάνω αλγόριθμο σε ένα πλήρως κατανεμημένο συνεργατικό σύστημα ανίχνευσης εισβολών. Τα πειράματα τους έδειξαν ότι αυτή η προσέγγιση μπορεί να πετύχει σημαντική μείωση του αριθμού των λάθος συναγερμών που παράγονται τοπικά σε ένα IDS.

Ο Junho Hong και ο Chen-Ching Liu μελέτησαν την χρήση ενός συνεργατικού δικτύου ανίχνευσης εισβολών σε ένα δίκτυο από έξυπνες ηλεκτρονικές συσκευές [11]. Οι συγκεκριμένες ηλεκτρονικές συσκευές έχουν την δυνατότητα να παρακολουθούν και να ανιχνεύουν ανωμαλίες και μη φυσιολογικές συμπεριφορές στο δίκτυο που τις φιλοξενεί. Επίσης έχουν την δυνατότητα να συνεργάζονται με τις άλλες γειτονικές συσκευές με σκοπό να παίρνουν ακριβείς αποφάσεις και να βρίσκουν την προέλευση πιθανών επιθέσεων. Στο συγκεκριμένο πείραμα χρησιμοποιήθηκε ένα κοινό ενσωματωμένο σύστημα μέτρησης της απόδοσης του συστήματος ανίχνευσης εισβολών μέσω λειτουργιών προστασίας ισχύος σε ένα δίκτυο παροχής ηλεκτρικού ρεύματος. Τα αποτελέσματα έδειξαν ότι το δίκτυο των ηλεκτρονικών συσκευών δουλεύει με ακρίβεια και αποτελεσματικότητα χωρίς να διατρέχει κίνδυνο υπερφόρτωσης ή πρόβλημα με τις αποστάσεις μεταξύ των συσκευών.

Οι Richeng Jin, Xiaofan He και Huaiyu Dai στη μελέτη τους για τον συμβιβασμό μεταξύ ιδιωτικότητας και χρησιμότητας σε ένα συνεργατικό σύστημα ανίχνευσης εισβολών, δημιούργησαν ένα παίγνιο για να προσεγγίσουν το θέμα μέσω της θεωρίας παιγνίων [14]. Προτείνουν ένα παίγνιο 2 επιπέδων με έναν ηγέτη και πολλούς ακόλουθους. Βάσει της θεωρητικής ανάλυσης τους είναι εφικτή η μοντελοποίηση των αναμενόμενων συμπεριφορών του επιτιθέμενου και του συστήματος ανίχνευσης εισβολών καθώς και η παραγωγή μίας καμπύλης χρησιμότητας-ιδιωτικότητας. Επιπρόσθετα, αποδεικνύεται η ύπαρξη της ισορροπίας του Nash και προτείνεται ένας ασύγχρονος δυναμικός αλγόριθμος για τον υπολο-

γισμό των βέλτιστων στρατηγικών συνεργασίας των συστημάτων ανίχνευσης εισβολών. Στο τέλος τα αποτελέσματα της προσομοίωσης τους δικαιώνουν καθώς φαίνεται να επικυρώνουν την ανάλυση τους.

Η ομάδα των Wenjuan Li και Weizhi Meng δημιούργησαν ένα ευαίσθητο μοντέλο εμπιστοσύνης μέσα σε ένα δίκτυο από συστήματα ανίχνευσης εισβολών [15]. Για την δημιουργία αυτού του μοντέλου χρησιμοποίησαν τεχνικές μηχανικής μάθησης με σκοπό να αυτοματοποιήσουν την διαδικασία και να δώσουν σε κάθε σύστημα την δυνατότητα να εκτιμήσει το πόσο έμπιστο είναι ένα άλλο σύστημα στο δίκτυο. Για την εκτίμηση, συνέκριναν την απόδοση τριών διαφορετικών εποπτευόμενων classifiers στην εκχώρηση τιμών ευαισθησίας και εξέτασαν το μοντέλο εμπιστοσύνης τους κάτω από διαφορετικά σενάρια επίθεσης σε ένα πραγματικό ασύρματο δίκτυο αισθητήρων. Τα πειραματικά τους αποτελέσματα έδειξαν πως το μοντέλο εμπιστοσύνης που δημιούργησαν μπορεί να ενισχύσει την ακρίβεια ανίχνευσης κακόβουλων κόμβων στο δίκτυο και έχει καλύτερη απόδοση συγκρινόμενο με παρόμοια μοντέλα που έχουν υλοποιηθεί.

Η ομάδα των Zhiyuan Tan και Upasana T. Nagar προσεγγίζουν την χρήση των συστημάτων ανίχνευσης εισβολών από την πλευρά των “Big Data” [23]. Στην έρευνα τους εξηγούν πως ο μεγάλος όγκος δεδομένων που αποθηκεύονται στο cloud αλλάζει τα επιχειρηματικά μοντέλα στο σημερινό κόσμο. Αναφέρουν πως η ασφάλεια και η ιδιωτικότητα αυτών των δεδομένων είναι σημαντικό πρόβλημα για τους ιδιοκτήτες τους. Αναλύουν πως μία ενίσχυση στην ασφάλεια των cloud συστημάτων είναι απαραίτητη και πώς αυτό θα μπορούσε να επιτευχθεί μέσω της χρήσης ενός συνεργατικού συστήματος ανίχνευσης εισβολών χωρίς όμως να εμβαθύνουν ως προς το σύστημα αυτό κάθε αυτό.

Η ομάδα των Hong Liang και Yufei Ge παρουσίασε ένα συνεργατικό σύστημα ανίχνευσης εισβολών για cloud computing το οποίο χτίζει μία πολυεπίπεδη αρχιτεκτονική ανίχνευσης εισβολών με σκοπό να κάνει πιο ακριβείς και αποτελεσματικούς τους προσδιορισμούς των εισβολών [16]. Στην αρχιτεκτονική αυτή, οι ανιχνευτές προσφέρονται ως μια υπηρεσία και ενσωματώνουν μηχανική μάθηση για την δημιουργία των κανόνων ανίχνευσης του συστήματος. Επίσης, σχεδίασαν έναν μηχανισμό ανταλλαγής συναγερμών μεταξύ των συστημάτων στην περιοχή του cloud καθώς και έναν τρόπο κοινοποίησης της γνώσης για εισβολές και ύποπτες επιθέσεις. Τα πειραματικά τους αποτελέσματα απέδειξαν πως το σύστημα τους όντως ενισχύει την ασφάλεια όταν συμβαίνουν επιθέσεις δικτύου και εξασφαλίζουν ότι και οι πάροχοι υπηρεσιών cloud αλλά και οι χρήστες προστατεύονται ικανοποιητικά.

Ο Anderson Morais και η Ana Cavalli μελετώντας μια παραλλαγή ad hoc δικτύου, τα πλεγματικά δίκτυα (WMN, εκ του Wireless Mesh Networks), τα οποία είναι ιδιαίτερα ευάλωτα σε κακόβουλους κόμβους, λόγω των εγγενών χαρακτηριστικών τους, όπως η αποκεντρωμένη υποδομή και η υψηλή εξάρτηση της συνεργασίας των κόμβων, προτείνουν μία αρχιτεκτονική κατανεμημένων και συνεργατικών συστημάτων ανίχνευσης εισβολών [17]. Η αρχιτεκτονική αυτή ανιχνεύει επιθέσεις σε πραγματικό χρόνο μέσω της ανάλυσης της κίνησης και της δημιουργία αντίστοιχων ροών επικοινωνίας. Μία κατανεμημένη μηχανή ανίχνευσης εισβολών (DIDE) εφαρμόζει περιορισμούς σε αυτές τις ροές υπολογίζει μετρήσεις σχετικές με κακή συμπεριφορά. Τέλος ένας συνεργατικός μηχανισμός ομοφωνίας (CCM) προβαίνει στον έλεγχο των μετρήσεων κακής συμπεριφοράς χρησιμοποιώντας ένα προτεινόμενο σχήμα κατωφλίου, με σκοπό τον εντοπισμό της πηγής των εισβολών. Όλο το συνεργατικό σύστημα ανίχνευσης εισβολών υλοποιήθηκε σε εικονική πλατφόρμα πλεγματικού δικτύου. Τα πειραματικά αποτελέσματα έδειξαν πως η προτεινόμενη αρχιτεκτονική συστήματος ανιχνεύει αποτελεσματικά τις επιθέσεις κατασκευής μηνυμάτων με υψηλή ακρίβεια και χαμηλή κατανάλωση πόρων.

Οι Mauro Andreolini, Michele Colajanni και Mirco Marchetti εισάγουν μια νέα κατηγορία επιθέσεων όπου ένας εισβολέας χωρίζει μέσω κατακερματισμού το κακόβουλο φορτίο με τέτοιο τρόπο ώστε κανένα κομμάτι του να μην μπορεί να ανιχνευθεί ακόμα και από τα πιο σύγχρονα συστήματα ανίχνευσης εισβολών [1]. Στη συνέχεια προτείνουν μία πρωτότυπη λύση και την υλοποιούν ως επέκταση του συστήματος Snort. Τα πειραματικά αποτελέσματά τους επιβεβαίωσαν την αποτελεσματικότητα της προτεινόμενης λύσης για τα διάφορα πρωτόκολλα που προσφέρουν δικτυακή κινητικότητα.

Οι ÁineMac Dermott, Qi Shi και Kashif Kifayat ανέπτυξαν μία μέθοδο ανίχνευσης εισβολής σε περιβάλλον cloud, για την συνεργατική ανίχνευση εισβολών. Χρησιμοποίησαν τη θεωρία Dempster-Shafer για να συνυπολογίσουν τα ευρήματα όλων των οντοτήτων παρακολούθησης και να πάρουν την τελική απόφαση σχετικά με μία πιθανή επίθεση [5].

Τέλος ο Εμμανουήλ Βασιλομανολάκης και ο Matthias Krugl εφιστούν την προσοχή στην ανάγκη για μετάβαση από τα παραδοσιακά απομονωμένα IDS σε ένα μεγάλο και κατανεμημένο συνεργατικό δίκτυο IDS (CIDS) [24]. Παρουσιάζουν μια νέα προσέγγιση CIDS, η οποία είναι σε θέση να διαμοιράζει συναγερμούς μόνο στους αισθητήρες παρακολούθησης που επιτρέπεται να επικοινωνούν μεταξύ τους. Επιπλέον κατά την διανομή των δεδομένων υποστηρίζουν πως το σύστημα διασφαλίζει ότι προστατεύεται το απόρρητο των δεδομένων αυτών. Ο κώδικας που χρησιμοποίησαν είναι open source και το πρότζεκτ τους ονομάζεται SkipMon.

Από την επισκόπηση της σχετικής βιβλιογραφίας καθίσταται σαφές πως οι συνεργατικές δομές ανίχνευσης εισβολών σε δίκτυα μπορούν να επιφέρουν μία σημαντική βελτίωση στην αποτελεσματικότητα των αντίστοιχων μεμονωμένων συστημάτων. Παρά την πληθώρα του δημοσιευμένου έργου στην περιοχή, ελάχιστοι ερευνητές ασχολούνται με μία πολύ σημαντική παράμετρο στα συνεργατικά αυτά συστήματα η οποία είναι η παραβίαση της ιδιωτικότητας των δεδομένων χρήσης ή δικτυακής κίνησης των επιμέρους συνεργαζόμενων οργανισμών. Στην παρούσα πτυχιακή παρουσιάζεται μία από τις πρώτες ερευνητικές προσπάθειες ανάπτυξης ενός τέτοιου συστήματος, που θα είναι ικανό να παράγει χρήσιμη πληροφορία για τους συνεργαζόμενους κόμβους, επιτρέποντάς τους όμως την διασφάλιση της ιδιωτικότητας των δεδομένων τους.

3. ΕΡΓΑΛΕΙΑ, ΜΕΘΟΔΟΙ ΚΑΙ ΜΑΘΗΜΑΤΙΚΑ ΜΟΝΤΕΛΑ

3.1 Αλγόριθμος K-Medoids

Ο αλγόριθμος **k-medoids** [21] είναι ένας αλγόριθμος ομαδοποίησης που σχετίζεται με τους αλγόριθμους **k-means** και **medoidshift**. Αμφότεροι οι αλγόριθμοι **k-means** και **k-medoids** διασπούν το σύνολο των δεδομένων σε ομάδες και προσπαθούν να ελαχιστοποιήσουν την απόσταση μεταξύ των σημείων που ανήκουν σε κάθε ομάδα και ενός σημείου ορισμένου ως το κέντρο της ομάδας αυτής. Σε αντίθεση με τον αλγόριθμο k-means, ο k-medoids επιλέγει σημεία δεδομένων (αντικείμενα) ως κέντρα και μετράει τις αποστάσεις με μία γενικοποίηση της απόστασης **Manhattan** και όχι με τη **νόρμα** l_2.

$$dist_{Manhattan} = \sum_{i=1}^{k} |x_i - y_i| \tag{3.1}$$

Ο **k-medoids** είναι μια κλασική τεχνική διαχωρισμού της ομαδοποίησης(PAM) που ομαδοποιεί το σύνολο δεδομένων **n** αντικειμένων σε **k** ομάδες οι οποίες είναι ορισμένες εκ των προτέρων (δεδομένες).

Είναι πιο ανθεκτικός στο θόρυβο και στις ακραίες τιμές σε σύγκριση με τον **k-means**. Αυτό επιτυγχάνεται ελαχιστοποιώντας ένα άθροισμα ζευγαρωμένων (pairwise) ανισοτήτων, αντί για ένα άθροισμα τετραγωνικών ευκλείδιων αποστάσεων.

Συνοπτικά τα βήματα του αλγορίθμου:

- Αυθαίρετη επιλογή των k objects σαν τα αρχικά medoids των επιμέρους ομάδων
- Επαναληπτική διαδικασία με τα εξής επιμέρους βήματα :
  - Τοποθέτηση κάθε object από τα υπόλοιπα στο κοντινότερο medoid
  - Για κάθε medoid o_j και για κάθε object o_{random} που ανήκει σε αυτό :
    * Υπολογισμός του συνολικού κόστους S της ανταλλαγής του o_j με το o_{random}
    * Εάν το $S < 0$ αντικατάσταση του o_j με το o_{random}
- Η επαναληπτική διαδικασία ολοκληρώνεται όταν δεν συμβαίνει καμία αλλαγή

3.1.1 Αλγόριθμοι ομαδοποίησης

3.1.1.1 K-means

- Επιλογή k τυχαίων objects που αποτελούν τα κέντρα των αρχικών k clusters. Τα objects είναι τα κέντρα τους.
- Υπολογισμός των αποστάσεων όλων των objects από το κέντρο κάθε cluster και τοποθέτησή τους στο cluster από το οποίο έχουν την μικρότερη απόσταση. Χρησιμοποιείται μία συνάρτηση simillarity η οποία δίνει την απόσταση του κάθε object από ένα cluster βάσει της απόστασής του από το κέντρο του cluster.
- Υπολογισμός εκ νέου των κέντρων των clusters. Επανακατανομή των objects στα clusters.
- Η διαδικασία ολοκληρώνεται (συγκλίνει), όταν παύουν να προκύπτουν αλλαγές στα cluster.

Ο αλγόριθμος χρησιμοποιείται όταν μπορεί να ορισθεί το κέντρο ενός cluster και δεν εφαρμόζεται όταν για παράδειγμα οι τιμές των δεδομένων αποτελούνται από categorical attributes. Ο αλγόριθμος δεν ευνοεί καθόλου clusters με μη-κυρτά σχήματα ή με μεγάλες διαφορές στο μέγεθος. Επιπλέον δεν αποδίδει καλά όταν υπάρχει έντονο το φαινόμενο των outlliers ή των noisy data, καθώς επηρεάζεται σημαντικά η μέση τιμή.

3.1.1.2 Medoids

Η πιο κοινή υλοποίηση της ομαδοποίησης k-medoid είναι ο αλγόριθμος καταμερισμού μεταξύ Medoids ή αλλιώς Συσταδοποίηση Διαμέρισης (Partitioning Around Medoids-PAM). Ο PAM χρησιμοποιεί μια άπληστη αναζήτηση (greedy) η οποία μπορεί να μην βρει τη βέλτιστη λύση, αλλά είναι ταχύτερη από την εξαντλητική αναζήτηση (exhaustive).

Λειτουργεί ως εξής:

1. Αρχικοποίηση: επιλογή τυχαίων **k** από τα **n** σημεία δεδομένων ως medoids.
2. Ανάθεση του κάθε σημείου δεδομένων στο πλησιέστερο medoid.
3. Υπολογισμός του συνολικό κόστος TC_{ih} για όλα τα ζεύγη των αντικειμένων O_i, O_h όπου το O_i είναι το τρέχον επιλεγμένο αντικείμενο και το O_h είναι ένα μη επιλεγμένο αντικείμενο.

4. Όσο το κόστος της διαμόρφωσης μειώνεται:
    i Για κάθε medoid **m**, για κάθε μη-medoid σημείο δεδομένων **o**:
        i Αντάλλαγη **m** και **o**, επανυπολογισμός του κόστους (το άθροισμα των αποστάσεων των σημείων προς το medoid τους)
        ii Εάν το συνολικό κόστος της διαμόρφωσης αυξήθηκε στο προηγούμενο βήμα, αναίρεση της εναλλαγής

Αλγόριθμοι εκτός από τον PAM έχουν επίσης προταθεί στη βιβλιογραφία, συμπεριλαμβανομένης της ακόλουθης μεθόδου επανάληψης **Voronoi**:

1. Επιλογή αρχικών medoids
2. Ελάττωση όσο το κόστος μειώνεται:
    i Σε κάθε cluster, ορισμός medoid ως το σημείο που ελαχιστοποιεί το άθροισμα των αποστάσεων μέσα στο cluster.
    ii Ανάθεση εκ νέου κάθε σημείου στο cluster που ορίζεται από το πλησιέστερο medoid που καθορίστηκε στο προηγούμενο βήμα.

3.1.2 Χρήση τεχνικών στην προτεινόμενη μεθοδολογία

Σε ορισμένες τυπικές καταστάσεις, ο αλγόριθμος k-medoids επιδεικνύει καλύτερη απόδοση από τον αλγόριθμο k-means. Το πιο χρονοβόρο μέρος του αλγορίθμου k-medoids είναι ο υπολογισμός των αποστάσεων μεταξύ αντικειμένων. Εάν εφαρμόζεται τετραγωνική προεπεξεργασία και αποθήκευση, ο πίνακας αποστάσεων μπορεί να είναι προϋπολογισμένος ώστε να επιτευχθεί αποτελεσματική επιτάχυνση.

Συμπερασματικά ορίζεται ως **ομαδοποίηση (clustering)** η οργάνωση μιας συλλογής από αντικείμενα-στοιχεία (objects) σε ομάδες (clusters) με βάση κάποιο μέτρο ομοιότητας και ως **medoid** το αντικείμενο ενός σετ του οποίου η μέση ανομοιογένεια προς όλα τα αντικείμενα του συμπλέγματος είναι ελάχιστη. Δηλαδή πιο απλά είναι ένα σημείο που βρίσκεται πιο κεντρικά στο σετ (μεσοειδες).

Σημαντικός παράγοντας για την επιλογή του αλγορίθμου **k-medoids**, για την διενέργεια του clustering των συναγερμών αποτέλεσε η χρήση των ίδιων των σημείων ως κέντρα των clusters. Αυτό επιτρέπει τον υπολογισμό και την κρυπτογράφηση των αποστάσεων μεταξύ

όλων των συνδυασμών των σημείων, εκ των προτέρων, και την χρήση τους όταν αυτό είναι επιθυμητό στην συνέχεια.

3.2 Ομομορφική Κρυπτογράφηση

Η Ομομορφική κρυπτογράφηση είναι μία μορφή κρυπτογράφησης που επιτρέπει την διενέργεια υπολογισμών στο κρυπτοκείμενο χωρίς να είναι απαραίτητη η αποκρυπτογράφησή του. Το αποτέλεσμα των πράξεων επιστρέφεται ως κρυπτογραφημένο αποτέλεσμα, το οποίο αν αποκρυπτογραφηθεί είναι το ίδιο με το αποτέλεσμα της ισοδύναμης πράξης πάνω στα αντίστοιχα απλά κείμενα.

Υπάρχουν πολλές μορφές μερικώς ομομορφικών κρυπτοσυστημάτων που επιτρέπουν την εφαρμογή κάποιων συγκεκριμένων πράξεων (πρόσθεσης και πολλαπλασιασμού). Πολλά κρυπτοσυστήματα με ομομορφικές ιδιότητες όπως ο αλγόριθμος RSA, ο αλγόριθμος Paillier και ο αλγόριθμος ElGamal είναι μερικώς ομομορφικά. Οι μέθοδοι που επιτρέπουν την διενέργεια οποιασδήποτε πράξης ονομάζονται πλήρως ομομορφικοί αλγόριθμοι. Λόγω κάποιων σημαντικών μειονεκτημάτων η ολική ομομορφική κρυπτογράφηση δεν είναι πολύ πρακτική στη χρήση της. Σημαντικότερα μειονεκτήματα είναι ο χρόνος επεξεργασίας και η πολυπλοκότητα της υλοποίησης.

Έχει αποδειχθεί πρόσφατα ότι ένα ολικώς ομομορφικό κρυπτοσύστημα είναι δυνατό να υλοποιηθεί, αλλα ακομα δεν υπάρχει κάποια αποδοτική υλοποίηση. Το 2009 ο Craig Gentry, παρουσίασε [8, 7] ότι ένα πλήρως ομομορφικό κρυπτοσύστημα είναι υλοποιήσιμο. Το κρυπτοσυστημα που παρουσίασε ο Gentry παράγει ένα πολύ μεγάλο κρυπτοκείμενο σε σύγκριση με το απλό κείμενο. Επίσης το γεγονός ότι είναι βασισμένο στο πλέγμα κάνει την υλοποίηση πολύπλοκη και οι πράξεις στο πεδίο των κρυπτοκειμένων είναι ιδιαίτερα απαιτητικές σε πόρους και χρόνο.

Στην συνέχεια έχουν προταθεί διάφορες εναλλακτικές υλοποιήσεις του αλγορίθμου του Gentry [9, 2, 22], οι οποίες βελτίωσαν σημαντικά τις επιδόσεις του αλγορίθμου. Ακόμα όμως και μετά από αυτές τις βελτιώσεις δεν υπάρχει καμία υλοποίηση η οποία να είναι κατάλληλη για χρήση στην λύση πρακτικών προβλημάτων σε λογικούς χρόνους.

3.2.1 Μερική ομομορφική κρυπτογράφηση

Ένα κρυπτοσύστημα ονομάζεται μερικώς ομομορφικό όταν μπορεί να κάνει είτε ομομορφική πρόσθεση είτε ομομορφικό πολλαπλασιασμό, αλλα όχι και τα δύο. Κάποια παραδείγματα τέτοιων ομομορφικών κρυπτοσυστημάτων ειναι:

- Ο πολλαπλασιαστικός ομομορφισμος RSA [19]
- Ο πολλαπλασιαστικός ομομορφισμός ElGamal [6]
- Ο προσθετικός ομομορφισμός Paillier [18]
- Ο ομομορφισμος Goldwasser-Micali [10]
- Ο ομομορφισμος Benaloh [3]

3.2.2 Ολική ομομορφική κρυπτογράφηση

Ένα κρυπτοσύστημα λογίζεται ως πλήρως ομομορφικό αν μπορεί να υλοποιήσει ομομορφικά και την πράξη της πρόσθεσης αλλά και αυτή του πολλαπλασιασμού. Το μοναδικό τέτοιο κρυπτοσύστημα είναι αυτό του Gentry που περιγράφηκε παραπάνω. Το σχήμα του Gentry βασίζεται σε ένα πολύπλοκο πλέγμα (mesh of ideal lattices) για την αναπαράσταση των κλειδιών και του κρυπτοκειμένου. Το ιδιωτικό κλειδί του συστήματος περιέχει έναν τυχαίο πίνακα **V** και έναν πίνακα **W** τέτοιους ώστε να ικανοποιούν την σχέση:

$$V \times W \equiv c \ \text{mod}\ f(x) \tag{3.2}$$

ώπου **c** είναι μία σταθερά.

Το δημόσιο κλειδί **B**, είναι η απλή ερμιτιανή μορφή του **V**.

3.2.3 Μειονεκτήματα

Αν και τα πλεονεκτήματα ενός ομομορφικού κρυπτοσυστήματος είναι πραγματικά σπουδαία, δεν προκύπτουν χωρίς αξιοσημείωτα μειονεκτήματα. Ένα από τα μεγαλύτερα μειονεκτήματα είναι η πολυπλοκότητα των ομομορφικών συστημάτων. Στα μερικώς ομομορφικά κρυπτοσυστήματα, δεν υπάρχει μεγάλος φόρτος επιβάρυνσης του υπολογιστικού συστήματος, τουλάχιστον για τους αλγόριθμους που αναφέρθηκαν παραπάνω. Ωστόσο,

η πλήρως ομομορφική κρυπτογράφηση απαιτεί κρυπτοσύστημα πλέγματος το οποίο είναι σημαντικά πιο πολύπλοκο.

Η υλοποίηση ενός τέτοιου κρυπτοσυστήματος ακόμα και για βασικές υπολογιστικές πράξεις απαιτεί αρκετά πιο περίπλοκους υπολογισμούς και παράγει υπερβολικά μεγάλα κρυπτοκείμενα. Άμα χρησιμοποιηθούν οι βασικές αρχές ασφάλειας, τα κρυπτοκείμενα που παράγονται είναι της τάξης των 128MB και το δημόσιο κλειδί της τάξης των 128PB. Κρυπτοκείμενα και δημόσια κλειδιά τέτοιου μεγέθους δεν είναι καθόλου πρακτικά. Ακόμα κι αν ελαχιστοποιήσουμε τις παραμέτρους ασφάλειας σε σημείο που ο ομομορφισμος να μην είναι πια δυνατός, το μέγεθος του κλειδιού θα είναι ακόμη της τάξης αρκετών GB, με την κρυπτογράφηση ενός απλού bit να χρειάζεται εώς και 30 λεπτά.

Ένα ακομη δυνητικό μειονέκτημα των ομομορφικών κρυπτοσυστημάτων είναι όι σε κάποιες περιπτώσεις, είναι εκτεθειμένα σε κακόβουλο λογισμικό (malware). Για παράδειγμα στην περίπτωση μιας ηλεκτρονικής ψηφοφορίας με ομομορφική πρόσθεση των ψήφων, είναι πιθανό ένα κακόβουλο λογισμικό να μπορούσε να επηρεάσει τις ψήφους πριν αυτές υποβληθούν.

3.3 Paillier

Το κρυπτοσύστημα Paillier, που εφευρέθηκε και πήρε το όνομα του από τον Pascal Paillier, είναι ένας πιθανοτικός(probabilistic) ασύμμετρος αλγόριθμος κρυπτογράφησης δημοσίου κλειδιού.

Στην ουσία είναι ένα ομομορφικό κρυπτοσύστημα πρόσθεσης, αυτό σημαίνει πως δοθέντος του δημοσίου κλειδιού και δυο κρυπτογραφημένων αριθμών **m1** και **m2**, κάποιος μπορεί να υπολογίσει το κρυπτογραφημένο αποτέλεσμα της πρόσθεσης τους **m1 + m2**.

Η λειτουργία του αλγορίθμου βασίζεται σε τρεις διαφορετικές διαδικασίες, την παραγωγή των κλειδιών, την κρυπτογράφηση και την αποκρυπτογράφηση, οι οποίες αναλύονται στην συνέχεια.

3.3.1 Δημιουργία κλειδιών

1. Ο κόμβος επιλέγει 2 μεγάλους πρώτους αριθμούς **p** και **q** τυχαία και ανεξάρτητα τον έναν από τον άλλον έτσι ώστε :

$$GCD(p * q, (p-1) * (q-1)) = 1 \tag{3.3}$$

   Αυτή η ιδιότητα είναι εξασφαλισμένη εάν και οι δύο αριθμοί έχουν το ίδιο μήκος.

2. Υπολογίζονται οι τιμές n, λ :

$$n = p * q \tag{3.4}$$

$$\lambda = lcm(p-1, q-1) \tag{3.5}$$

3. Επιλέγει τυχαίο ακέραιο **g** όπου $g \in \mathbb{Z}^*_{n^2}$

4. Υπολογίζει τον μ ως τον αντίστροφο ως προς $mod n$ του αριθμού $L(g^\lambda \cdot n^2)$:

$$\mu = (L(g^\lambda \cdot n^2))^{-1} mod n \tag{3.6}$$

   όπου συνάρτηση L ορίζεται η $L(x) = \frac{x-1}{n}$.

   Το δημόσιο κλειδί κρυπτογράφησης είναι (n,g), ενώ το ιδιωτικό κλειδί κρυπτογράφησης είναι (λ,μ).

   Αν χρησιμοποιούμε p, q ισοδύναμου μήκους, μια απλούστερη παραλλαγή των παραπάνω βημάτων παραγωγής κλειδιών θα ήταν να ορίσουμε $g = n+1, \lambda = \varphi(n)$, $g = n+1, \lambda = \varphi(n)$, και $\mu = \varphi(n)^{-1} \bmod n$, $\mu = \varphi(n)^{-1} \bmod n$, όπου $\varphi(n) = (p-1)(q-1)$.

3.3.2 Κρυπτογράφηση

1. Έστω **m** το κρυπτογραφημένο μήνυμα, όπου $0 \leq m < n$

2. Επιλογή τυχαίου r όπου $0 \leq r < n$

3. Υπολογισμός κρυπτοκειμένου με την φόρμουλα : $c = g^m \cdot r^n mod n^2$

3.3.3 Αποκρυπτογράφηση

1. Έστω **c** το κρυπτοκείμενο για αποκρυπτογράφηση, όπου $c \in \mathbb{Z}_{n^2}^*$
2. Υπολογισμός απλού κειμένου με την φόρμουλα: $m = L(c^\lambda \bmod n^2) \cdot \mu \bmod n$

3.3.4 Ομομορφικές ιδιότητες

Ένα αξιοσημείωτο χαρακτηριστικό του κρυπτοσυστήματος Paillier είναι οι ομομορφικές του ιδιότητες καθώς και η μη-ντετερμινιστική κρυπτογράφηση του. Επειδή η λειτουργία κρυπτογράφησης είναι προσθετικά ομομορφική, μπορούν να περιγραφούν οι ακόλουθες ταυτότητες:

- Ομομορφική πρόσθεση απλών κειμένων

  Είναι εφικτή η πρόσθεση δύο κρυπτογραφημένων αριθμών. Το προϊόν δύο κρυπτοκειμένων θα αποκρυπτογραφηθεί στο άθροισμα των αντίστοιχων απλών κειμένων τους.

$$D(E(m_1, r_1) \cdot E(m_2, r_2) \bmod n^2) = m_1 + m_2 \bmod n \tag{3.7}$$

- Ομομορφική πρόσθεση κρυπτοκειμένου και απλού κειμένου

  Είναι εφικτή η πρόσθεση ενός γνωστού αριθμού με έναν κρυπτογραφημένο αριθμό. Το προϊόν ενός κρυπτοκείμενου και της δύναμης με βάση g και εκθέτη ένα απλό κείμενο θα αποκρυπτογραφηθεί στο άθροισμα των αντίστοιχων απλών κειμένων.

$$D(E(m_1, r_1) \cdot g^{m_2} \bmod n^2) = m_1 + m_2 \bmod n \tag{3.8}$$

- Ομομορφικός πολλαπλασιασμός κρυπτοκειμένου και απλού κειμένου

  Είναι εφικτός ο πολλαπλασιασμός ενός κρυπτογραφημένου αριθμού με έναν γνωστό αριθμό. Ένα κρυπτογραφημένο απλό κείμενο υψωμένο στη δύναμη ενός άλλου απλού κειμένου θα αποκρυπτογραφηθεί στο προϊόν των δύο απλών κειμένων.

$$D(E(m_1, r_1)^{m_2} \bmod n^2) = m_1 m_2 \bmod n \tag{3.9}$$

Γενικότερα, ένα κρυπτογραφημένο απλό κείμενο που υψώνεται σε μια σταθερά k θα αποκρυπτογραφηθεί στο προϊόν του απλού κειμένου και της σταθεράς,

$$D(E(m_1, r_1)^k \mathsf{mod} n^2) = km_1 \mathsf{mod} n \tag{3.10}$$

Ωστόσο, δοθέντων των κρυπτογραφημάτων Paillier δύο μηνυμάτων δεν υπάρχει γνωστός τρόπος για να υπολογιστεί η κρυπτογράφηση του προϊόντος αυτών των μηνυμάτων χωρίς να γνωρίζουμε το ιδιωτικό κλειδί. Αυτή η αδυναμία ουσιαστικά καθιστά τον αλγόριθμο Paillier μη πλήρως ομομορφικό.

Τέλος μία άλλη ενδιαφέρουσα ιδιότητα του αλγορίθμου είναι η μη ντετερμινιστική κρυπτογράφηση. Ο ακέραιος αριθμός r που χρησιμοποιείται κατά την κρυπτογράφηση επιλέγεται τυχαία, με αποτέλεσμα δύο διαδοχικές κρυπτογραφήσεις του ίδιου απλού κειμένου να παράγουν διαφορετικά κρυπτοκείμενα. Η συγκεκριμένη ιδιότητα είναι σημαντική, καθώς αποτρέπει όποιον έχει πρόσβαση στα κρυπτοκείμενα δύο μεταβλητών με τις ίδιες τιμές να αντιληφθεί την συγκεκριμένη ισότητα.

3.3.5 Εφαρμογές

Ηλεκτρονική ψηφοφορία

Οι παραπάνω ομομορφικές ιδιότητες μπορούν να χρησιμοποιηθούν από ασφαλή ηλεκτρονικά συστήματα ψηφοφορίας. Για παράδειγμα, έστω μια απλή ψηφοφορία ”υπέρ-κατά”. Έστω k ο αριθμός των ψηφοφόρων που ψηφίζουν είτε **1 (υπέρ)** είτε **0 (κατά)**. Κάθε ψηφοφόρος κρυπτογραφεί την επιλογή του. Ο εκλογικός υπεύθυνος παίρνει το προϊόν των **y** κρυπτογραφημένων ψήφων και στη συνέχεια αποκρυπτογραφεί το αποτέλεσμα και αποκτά την τιμή **n**, η οποία και είναι το άθροισμα όλων των ψήφων. Ο εκλογικός υπεύθυνος τότε γνωρίζει ότι **n** άνθρωποι ψήφισαν υπερ και **m-n** άνθρωποι ψήφισαν κατά. Το τυχαίο **r** εξασφαλίζει ότι δύο ταυτόσημες ψήφοι έχουν αμελητέα πιθανότητα να κρυπτογραφούν στην ίδια τιμή, εξασφαλίζοντας έτσι την ιδιωτικότητα των ψηφοφόρων.

Ηλεκτρονικά μετρητά

Μία άλλη λειτουργία της ομομορφικής κρυπτογράφησης είναι η έννοια του αυτοαποκλεισμού. Αυτή είναι η δυνατότητα αλλαγής ενός κρυπτοκειμένου σε άλλο χωρίς να αλλάζει το περιεχόμενο της αποκρυπτογράφησης του. Αυτό μπορεί να εφαρμοστεί στην εξέλιξη του ecash, ενός ηλεκτρονικου συστήματος μετρητών, μια προσπάθεια που αρχικά πρωτοστάτησε ο David Chaum. Μέσω αυτής της εφαρμογής της ομομορφικής κρυπτογράφησης

θα μπορεί καποιος να πληρώνει για κάτι στο διαδίκτυο χωρίς ο πωλητής να χρειάζεται να γνωρίζει τον αριθμό της πιστωτικής του κάρτας και, συνεπώς, την ταυτότητά του.

Ο στόχος τόσο στο ηλεκτρονικό χρήμα όσο και στην ηλεκτρονική ψηφοφορία είναι να διασφαλιστεί ότι το ηλεκτρονικό κέρμα ή αντίστοιχα η ηλεκτρονική ψηφοφορία είναι έγκυρα, ενώ παράλληλα να μην αποκαλύπτεται η ταυτότητα του ατόμου με το οποίο συσχετίζεται αυτή τη στιγμή.

Βιοϊατρική τεχνολογία

Όσο κι αν φαίνεται περίεργο, η ομομορφική κρυπτογράφηση θα μπορούσε να εφαρμοστεί και στον τομέα της βιοϊατρικης τεχνολογίας. Είναι γνωστό πως το ιατρικό απόρρητο και η αυστηρή χρήση για λόγους ασφάλειας των αλγορίθμων των βιοϊατρικών μηχανημάτων, απαγορεύει από μία τρίτη οντότητα την ομαδοποίηση μεγάλων πακέτων δεδομένων ασθενών. Σε μια προσπάθεια εξαγωγής χρήσιμων συμπερασμάτων θα μπορούσαμε να χρησιμοποιήσουμε μία τρίτη οντότητα με μεγάλη υπολογιστική ισχύ για ομαδοποίηση ιατρικών δεδομένων, για παράδειγμα εικόνες MRI.

Έτσι θα είχαμε απίστευτα μεγάλο αριθμό μαγνητικών τομογραφιών χωρισμένων σε ομάδες και θα ήταν ευκολότερη για παράδειγμα η ”χαρτογραφηση” των διαφορετικών περιοχών του εγκεφάλου ή σε άλλη περίπτωση η ανίχνευση των πιο επικίνδυνων περιοχών εκδήλωσης ενός καρκινικού όγκου. Όλα αυτά χωρίς να εκθέταμε σε αυτήν την τρίτη οντότητα τα ιδιωτικά δεδομένα κάθε ασθενή.

Βιοπληροφορική

Αντίστοιχα, στον τομέα της βιοπληροφορικής, όπου ένας πλούτος προσωπικών γονιδιωματικών δεδομένων γίνεται διαθέσιμος χάρη στην επιστημονική πρόοδο πάνω στην αλληλουχία του ανθρώπινου γονιδιώματος και τεχνικών συναρμολόγησης γονιδίων, νοσοκομεία, ερευνητικά ιδρύματα, κλινικές και εταιρείες χειρισμού ανθρώπινου γονιδιωματικού υλικού και άλλων ευαίσθητων δεδομένων για την υγεία αντιμετωπίζουν το κοινό πρόβλημα ασφαλής αποθήκευσης και ομαδοποίησης για αλληλοσυσχετισμο μεγάλων ποσοτήτων δεδομένων. Ποιός θα εμπιστευτεί αυτά τα ευαίσθητα δεδομένα σε μια τρίτη οντότητα για επεξεργασία χωρίς να ανησυχεί για την παραβίαση τους; Με την χρήση της ομομορφικής κρυπτογράφησης αυτό το μεγάλο εμπόδιο ξεπερνιέται.

3.4 Συστήματα Ανίχνευσης Εισβολων

Εισβολή είναι η μη-εξουσιοδοτημένη πρόσβαση/δραστηριότητα ή προσπάθεια πρόσβασης σε έναν υπολογιστή ή ένα πληροφοριακό σύστημα. Ένα σύστημα ανίχνευση εισβολών η εν συντομία ”IDS” παρακολουθεί έναν υπολογιστή σε πραγματικό χρόνο για ύποπτη δραστηριότητα ή πραγματική εισβολή από κάποιον άλλο άνθρωπο ή υπολογιστή. Το σύστημα ανιχνεύει μη εξουσιοδοτημένους χρήστες που προσπαθούν να αποκτήσουν πρόσβαση στον υπολογιστή χρησιμοποιώντας μια μέθοδο σύγκρισης της συμπεριφοράς του χρήστη με το γενικό προφίλ που έχει χτίσει για τον εξουσιοδοτημένο χρήστη του συστήματος. Ανιχνεύει περιστατικά που δείχνουν μια μη-εξουσιοδοτημένη εισβολή στο υπολογιστικό σύστημα, ειδοποιεί μια λειτουργία ελέγχου για μη-εξουσιοδοτημένους χρήστες και περιστατικά ώστε αυτή αυτόματα να αντιμετωπίζει την εκάστοτε περίπτωση. Το προφίλ του εξουσιοδοτημένου χρήστη δημιουργείται δυναμικά για κάθε χρήστη υπολογιστή, όταν αυτός συνδέεται πρώτη φορά στον υπολογιστή και έπειτα ενημερώνεται κατά τις επόμενες συνδέσεις. Έτσι, συγκρίνοντας την συμπεριφορά του εκάστοτε χρήστη με το προφίλ που έχει χτίσει για τον εξουσιοδοτημένο μειώνει σημαντικά τους λάθος συναγερμούς.

Υπάρχει ένα ευρύ φάσμα συστημάτων ανίχνευσης εισβολών (IDS), που κυμαίνεται από λογισμικό προστασίας από ιούς έως ιεραρχικά συστήματα που παρακολουθούν την κυκλοφορία ολόκληρου του δικτύου. Η πιο κοινή ταξινόμηση χωρίζει τα συστήματα σε συστήματα ανίχνευσης εισβολής σε δίκτυο (NIDS) και σε συστήματα ανίχνευσης εισβολής που είναι βασισμένα σε έναν κεντρικό υπολογιστή (HIDS). Ένα σύστημα που παρακολουθεί σημαντικά αρχεία λειτουργικού συστήματος είναι ένα παράδειγμα ενός HIDS, ενώ ένα σύστημα που αναλύει την εισερχόμενη κίνηση δικτύου είναι ένα παράδειγμα ενός NIDS.

Είναι επίσης δυνατή η ταξινόμηση των IDS με βάση τον τρόπο ανίχνευσης: οι πιο γνωστές παραλλαγές είναι η ανίχνευση που στηρίζεται στην υπογραφή (signature-based) η οποία λειτουργεί αναγνωρίζοντας κάποια επικίνδυνα πρότυπα, όπως το κακόβουλο λογισμικό και η ανίχνευση που στηρίζεται σε ανωμαλίες (anomaly-based) η οποία βασίζεται στην μηχανική μάθηση και ανιχνεύει αποκλίσεις από ένα μοντέλο ”καλής κυκλοφορίας”. Ορισμένα IDS έχουν τη δυνατότητα να ανταποκρίνονται στις εισβολές που εντοπίζουν. Τα συστήματα με δυνατότητες απόκρισης αναφέρονται συνήθως ως συστήματα πρόληψης εισβολής.

3.4.1 Σύγκριση IDS με τείχος προστασίας

Παρόλο που και τα δύο σχετίζονται με την ασφάλεια δικτύων, ένα IDS διαφέρει από ένα τείχος προστασίας στο ότι ένα τείχος προστασίας βλέπει προς τα έξω για εισβολές, με σκοπό να τις εμποδίσει να συμβούν. Τα τείχη προστασίας περιορίζουν την πρόσβαση μεταξύ των δικτύων για να αποτρέψουν μια εισβολή και δεν δίνουν σήμα για μία επίθεση από το εσωτερικό του δικτύου. Ένα IDS περιγράφει μια υποτιθέμενη εισβολή όταν αυτή έχει ήδη πραγματοποιηθεί και παράγει έναν συναγερμό. Επίσης ένα IDS παρακολουθεί για επιθέσεις που προέρχονται από το εσωτερικό ενός συστήματος. Αυτό επιτυγχάνεται εξετάζοντας την επικοινωνία του δικτύου, προσδιορίζοντας τα ευρετικά και τα πρότυπα (συχνά γνωστά ως υπογραφές) των κοινών επιθέσεων ηλεκτρονικών υπολογιστών και λαμβάνοντας μέτρα για την ειδοποίηση των χρηστών. Ένα σύστημα που τερματίζει τις συνδέσεις ονομάζεται σύστημα πρόληψης εισβολής και είναι μια άλλη μορφή ενός τείχους προστασίας επιπέδου εφαρμογής.

Δικτυακά Συστήματα Ανίχνευσης Εισβολών

Τα δικτυακά συστήματα ανίχνευσης εισβολών (NIDS) τοποθετούνται σε στρατηγικά σημεία εντός του δικτύου για να παρακολουθούν την κίνηση από και προς όλες τις συσκευές του δικτύου. Ένα τέτοιο σύστημα πραγματοποιεί ανάλυση διερχόμενης κίνησης σε ολόκληρο το υποδίκτυο και συγκρίνει την κίνηση του υποδικτύου με μια βιβλιοθήκη από γνωστές επιθέσεις. Όταν ανιχνεύσει μία επίθεση ή περίεργη συμπεριφορά, στέλνει έναν συναγερμό στον διαχειριστή του συστήματος. Ιδανικά θα μπορούσε κάποιος να σαρώνει όλο το δίκτυο για εισβολές αλλά κάτι τέτοιο θα δημιουργούσε υπεφόρτωση του δικτύου. Μια πιθανή χρήση ενός NIDS θα μπορούσε να είναι η τοποθέτηση του στο υποδίκτυο και ο έλεγχος για πιθανές απόπειρες εισβολής στο τείχος προστασίας. Υπάρχουν NIDS που λειτουργούν online και άλλα που λειτουργούν και offline. Μερικά από τα πιο γνωστά NIDS συστήματα είναι τα OPNET και NetSim.

Συστήματα ανίχνευσης εισβολών στον εξυπηρετητή

Τα συστήματα ανίχνευσης εισβολών που βασίζονται στον εξυπηρετητή (HIDS) τρέχουν σε αυτόνομους εξυπηρετητές ή συσκευές στο δίκτυο. Ένα HIDS παρακολουθεί την εισερχόμενη και εξερχόμενη κίνηση πακέτων από την συσκευή μόνο και στέλνει συναγερμό στον χρήστη ή στον διαχειριστή αν διαγνώσει ύποπτη δραστηριότητα. Παίρνει ένα στιγμιότυπο των υπαρχόντων αρχείων συστήματος και τα συγκρίνει με το προηγούμενο στιγμιότυπο που έχει αποθηκεύσει. Αν τα σημαντικά αρχεία συστήματος έχουν διαγραφεί ή τροποποιηθεί, στέλνει έναν συναγερμό στον διαχειριστή για να το διερευνήσει περαιτέρω.

Ένα παράδειγμα χρήσης HIDS μπορεί να είναι σε mission-critical μηχανές, στις οποίες δεν αναμένεται να αλλάξει κάτι στις ρυθμίσεις τους.

Περιορισμοί

Όπως σε όλα τα συστήματα έτσι και στα συστήματα ανίχνευσης εισβολών υπάρχουν κάποιοι περιορισμοί που κάνουν την λειτουργία τους δυσκολότερη. Ο θόρυβος μπορεί ποικιλοτρόπως να μειώσει την αποτελεσματικότητα τους. Επίσης η δημιουργία κακών πακέτων από πιθανά bugs λογισμικού μπορεί να δημιουργήσει ψευδής συναγερμούς. Τις περισσότερες φορές εξάλλου οι συναγερμοί που αντιστοιχούν σε πραγματικές επιθέσεις είναι σημαντικά λιγότεροι από τους λάθος συναγερμούς και αυτό έχει ως αποτέλεσμα αρκετά συχνά οι πραγματικές επιθέσεις να περνάνε απαρατήρητες. Ακόμα, τα περισσότερα IDS δεν έχουν την δυνατότητα επεξεργασίας κρυπτογραφημένων πακέτων με επακόλουθο ένα κρυπτογραφημένο πακέτο δικτύου να μπορεί ανενόχλητο να διεισδύσει στο σύστημα. Το λογισμικό ανίχνευσης εισβολής παρέχει πληροφορίες βασιζόμενο στην διεύθυνσης δικτύου που συνδέεται με το πακέτο IP που αποστέλλεται στο δίκτυο. Αυτό είναι χρήσιμο εάν η διεύθυνση δικτύου που συνδέεται με το πακέτο IP είναι ακριβής, το οποίο δεν ισχύει πάντα.

3.4.2 Snort

Το **Snort** είναι ένα δωρεάν και open source σύστημα ανίχνευσης και παρεμπόδισης εισβολών που δημιουργήθηκε το 1998 απο τον Martin Roesch. Έχει την ικανότητα να εκτελεί ανάλυση κίνησης και καταγραφή πακέτων σε πραγματικό χρόνο ενός δικτύου IP. Το πρόγραμμα μπορεί επίσης να ανιχνεύσει επιθέσεις, συμπεριλαμβανομένων, των προσπαθειών αποτύπωσης λειτουργικού συστήματος, σημασιολογικών επιθέσεων διευθύνσεων URL, υπερχείλισης buffer, ανιχνευτών μπλοκ μηνυμάτων διακομιστών και σάρωσης ψεύτικων θυρών.

Το Snort μπορεί να ρυθμιστεί σε τρεις βασικές λειτουργίες: sniffer, καταγραφή πακέτων και ανίχνευση εισβολών δικτύου. Στη λειτουργία sniffer, το πρόγραμμα διαβάζει τα πακέτα δικτύου και τα εμφανίζει στην κονσόλα. Στη λειτουργία καταγραφής πακέτων, το πρόγραμμα καταγράφει τα πακέτα στο δίσκο. Στη λειτουργία ανίχνευσης εισβολών, το πρόγραμμα παρακολουθεί την κυκλοφορία του δικτύου και το αναλύει σε σχέση με ένα σύνολο κανόνων που ορίζει ο χρήστης. Το πρόγραμμα μπορεί να εκτελέσει στη συνέχεια μια συγκεκριμένη ενέργεια βάσει του ποια επίθεση έχει εντοπιστεί.

4. ΑΝΑΛΥΣΗ ΣΥΣΤΗΜΑΤΟΣ ΚΑΙ ΜΕΘΟΔΟΛΟΓΙΑ

4.1 Ανάλυση Συστήματος

4.1.1 Αρχιτεκτονική

Το προτεινόμενο σύστημα δίνει την δυνατότητα σε πολλαπλούς οργανισμούς να συνεργάζονται για την επεξεργασία των συναγερμών που παράγονται από τα συστήματα ανίχνευσης εισβολών τους με σκοπό την εξαγωγή χρήσιμων συμπερασμάτων. Συγκεκριμένα οι εκάστοτε οργανισμοί μπορούν να ομαδοποιήσουν τους συναγερμούς συνεργατικά και να εξάγουν χρησιμότερα συμπεράσματα για της επιθέσεις που εκτελούνται, ακόμα κι αν δεν έχουν πέσει ήδη θύματα. Το σύστημα αποτελείται από δύο υποσυστήματα, το υποσύστημα του Client το οποίο είναι εγκατεστημένο σε κάθε συμμετέχοντα οργανισμό και το υποσύστημα του Server το οποίο είναι εγκατεστημένο σε μια έμπιστη τρίτη οντότητα. Αυτά συνεργάζονται προκειμένου να εκτελέσουν την ομαδοποίηση των συναγερμών για ένα συγκεκριμένο χρονικό διάστημα.

Καθένα από τα υποσυστήματα των Clients είναι στενά συνδεδεμένο με ένα τοπικά εγκατεστημένο σύστημα ανίχνευσης εισβολών (IDS). Διαβάζει τους συναγερμούς που παράγονται από το IDS, τους κρυπτογραφεί και ύστερα τους τροφοδοτεί ως είσοδο στην συνολική διαδικασία ομαδοποίησης. Καθ'όλη την διαδικασία ομαδοποίησης, οι Clients υποστηρίζουν τον Server καθώς κάποιες από τις πράξεις που απαιτούνται δεν είναι εφικτό να πραγματοποιηθούν με τα κρυπτογραφημένα δεδομένα των συναγερμών. Τέλος, οι πληροφορίες για τις παραγόμενες ομάδες επιστρέφονται στους Clients και αυτοί με την σειρά τους τις αποκρυπτογραφούν με σκοπό να έχουν πρόσβαση σε αυτές.

Η κύρια επεξεργασία σχετικά με την ομαδοποίηση των συναγερμών, εκτελείται στο υποσύστημα του Server, το οποίο και είναι εγκατεστημένο σε μία έμπιστη τρίτη οντότητα. Αυτό δέχεται τους κρυπτογραφημένους συναγερμούς από τους Clients και εκτελεί ομαδοποίηση με τον αλγόριθμο k-medoids. Οποτεδήποτε κάποια πράξη καθίσταται αδύνατη λόγω της κρυπτογράφησης, ο Server ζητάει βοήθεια από κάποιον από τους Clients για να την πραγματοποιήσει. Τα αποτελέσματα της διαδικασίας είναι κρυπτογραφημένες πληροφορίες σχετικά με τις ομάδες συναγερμών που προκύπτουν και στη συνέχεια επιστρέφονται στους Clients.

4.1.2 Clients

Όλα τα υποσυστήματα των Clients μοιράζονται ένα κοινό ζευγάρι κλειδιών Paillier. Συλλέγουν τους συναγερμούς που παράγονται από τα τοπικά IDSs για ένα δεδομένο χρονικό διάστημα. Όσο παράγονται αυτοί οι συναγερμοί, τα υποσυστήματα των Clients εκτελούν τις εξής διεργασίες :

- Τους κρυπτογραφούν με το κοινό δημόσιο κλειδί Paillier.
- Υπολογίζουν τις αποστάσεις μεταξύ των τοπικά παραγόμενων συναγερμών.
- Κρυπτογραφούν τις υπολογισμένες αποστάσεις με το κοινό δημόσιο Paillier κλειδί.

Στο τέλος του δεδομένου χρονικού περιθωρίου, οι Clients στέλνουν στον Server τη λίστα με τους κρυπτογραφημένους συναγερμούς μαζί με τον άνω τριγωνικό πίνακα των κρυπτογραφημένων αποστάσεων μεταξύ των συναγερμών.

Όταν όλοι οι Clients στείλουν τα δεδομένα τους στον Server, μπαίνουν σε κατάσταση διαθεσιμότητας για να υποστηρίξουν τον Server όποτε αυτός τους το ζητήσει. Οποτεδήποτε ο Server χρειάζεται να εκτελέσει μια μη-εφικτή πράξη στα κρυπτογραφημένα δεδομένα, διαλέγει τυχαία έναν client για να του στείλει τα δεδομένα, με σκοπό να υλοποιήσει την πράξη. Ο συγκεκριμένος client αποκρυπτογραφεί τα δεδομένα με το κοινό ιδιωτικό κλειδί Paillier, εκτελεί την πράξη, κρυπτογραφεί το αποτέλεσμα με το κοινό δημόσιο κλειδί Paillier και το επιστρέφει στον Server.

Στο τέλος όταν η ομαδοποίηση έχει οριστικοποιηθεί, οι Clients λαμβάνουν τις προκύπτουσες πληροφορίες σχετικά με τις ομάδες από τον Server και τις αποκρυπτογραφούν χρησιμοποιώντας το κοινό ιδιωτικό κλειδί Paillier.

4.1.3 Server

Στο τέλος κάθε χρονικού παραθύρου ο Server περιμένει για όλους τους Clients να του στείλουν τα δεδομένα τους. Συγκεκριμένα, τους κρυπτογραφημένους συναγερμούς τους μαζί με τις κρυπτογραφημένες αποστάσεις των συναγερμών τους. Από την στιγμή που λαμβάνει τις πληροφορίες από όλους τους Clients, αρχίζει να υπολογίζει τις αποστάσεις μεταξύ των συναγερμών διαφορετικών Clients. Όταν τις υπολογίσει και αυτές, ο Server ”χτίζει” έναν πίνακα με τις αποστάσεις όλων των συναγερμών μεταξύ τους και ξεκινάει την εκτέλεση του αλγορίθμου k-medoids.

Ο Server κάνει χρήση των ομομορφικών ιδιοτήτων του αλγορίθμου Paillier, με σκοπό να εκτελέσει προσθέσεις μεταξύ κρυπτογραφημένων τιμών και προσθέσεις και πολλαπλασιασμούς μεταξύ μιας κρυπτογραφημένης και μίας μη κρυπτογραφημένης τιμής. Όποτε ο Server χρειάζεται να εκτελέσει οποιαδήποτε άλλη πράξη, ζητάει βοήθεια από έναν τυχαία επιλεγμένο Client, όπως αναφέρθηκε και προηγουμένως. Αυτή η προσέγγιση πρακτικά ξεπερνά τους περιορισμούς της μερικά ομομορφικής κρυπτογράφησης, αλλά από την άλλη πλευρά κοστίζει σε αποτελεσματικότητα και ιδιωτικότητα. Οι επιπτώσεις αυτής της προσέγγισης αναφέρονται στην ενότητα 5.

4.1.4 Work-flow

Η ροή εργασίας για ένα συγκεκριμένο χρονικό περιθώριο περιγράφεται στην παρακάτω υποενότητα. Ο αλγόριθμος K-medoids χρειάζεται ένα προκαθορισμένο αριθμό από Clusters για να λειτουργήσει. Αρχικά ο Server ορίζει αυτόν τον αριθμό ως k (αριθμός των clusters).

4.1.4.1 Οι Clients στέλνουν δεδομένα στον Server

Το πρώτο βήμα της διαδικασίας είναι η αποστολή των αρχικών δεδομένων από τους Clients στον Server. Κάθε Client πρέπει να μοιραστεί τα δεδομένα για του συναγερμούς που έχει συλλέξει το τοπικό IDS στο τελευταίο χρονικό διάστημα. Αυτά τα δεδομένα κρυπτογραφούνται με τον αλγόριθμο του Paillier πριν σταλθούν στον Server. Ο Server θα υπολογίσει τις αποστάσεις για όλα τα ζεύγη συναγερμών που θα μαζέψει από τους Clients. Ο ορισμός της απόστασης μεταξύ των συναγερμών παρουσιάζεται στην επόμενη ενότητα.

Καθώς αυτή είναι μία πολύ βαριά διεργασία, όσον αφορά την κατανάλωση υπολογιστικών πόρων, κάθε Client υπολογίζει τις αποστάσεις για τα ζεύγη των τοπικών συναγερμών που έχουν παραχθεί στο σύστημα του και είναι διαθέσιμοι σε μη κρυπτογραφημένη μορφή. Αυτό γλιτώνει τον Server από μία σημαντική ποσότητα πράξεων. Ειδικότερα, αν έχουμε n Clients, από τους οποίους ο καθένας έχει κατά προσέγγιση m_c συναγερμούς, τότε ο συνολικός αριθμός των αποστάσεων που πρέπει να υπολογιστούν είναι $calc_{tot}$:

$$calc_{tot} = \frac{n * m_c(n * m_c - 1)}{2} \tag{4.1}$$

Ο αριθμός των υπολογισμών που μπορούν να γίνουν τοπικά είναι $calc_{loc}$:

$$calc_{loc} = n * \frac{m_c(m_c - 1)}{2} \tag{4.2}$$

Ο λόγος αυτών των υπολογισμών είναι:

$$ratio = \frac{calc_{loc}}{calc_{tot}} = \frac{n * \frac{m_c(m_c-1)}{2}}{\frac{n*m_c(n*m_c-1)}{2}} = \frac{m_c - 1}{n * m_c - 1} \tag{4.3}$$

Οπότε για έναν σχετικά μικρό αριθμό Clients n και έναν μεγάλο αριθμό συναγερμών ανά Client m_c (το οποίο είναι και η πιο κοινή περίπτωση), αυτός ο λόγος είναι σχεδόν ίσος με $\frac{1}{n}$ το οποίο αποτελεί ένα σημαντικό ποσοστό των υπολογισμών.

Μετά το πρώτο βήμα, ο Server κρατά όλους τους συναγερμούς όλων των Clients σε κρυπτογραφημένη μορφή. Επιπλέον κρατάει τις αποστάσεις μεταξύ των συναγερμών που έχουν παραχθεί στον ίδιο Client.

4.1.4.2 Ο Server υπολογίζει τις υπόλοιπες αποστάσεις

Το δεύτερο βήμα που πρέπει να εκτελεστεί είναι ο υπολογισμός όλων των αποστάσεων μεταξύ των ζευγών συναγερμών που έχουν παραχθεί σε διαφορετικούς Clients. Ο υπολογισμός αυτών των αποστάσεων πρέπει να γίνει στον Server, και οι συναγερμοί να παραμείνουν κρυπτογραφημένοι. Όπως έχει αναφερθεί και στην υποενότητα 3.3, ο αλγόριθμος του Paillier χαρακτηρίζεται από κάποιες συγκεκριμένες ομομορφικές ιδιότητες οι οποίες καθιστούν δυνατή την εκτέλεση προσθέσεων μεταξύ κρυπτογραφημένων τιμών, και πολλαπλασιασμών και προσθέσεων μεταξύ μίας κρυπτογραφημένης και μίας μη κρυπτογραφημένης τιμής. Ένα μέρος του υπολογισμού των αποστάσεων των συναγερμών απαιτεί πολλαπλασιασμούς μεταξύ κρυπτογραφημένων τιμών. Αυτοί οι υπολογισμοί υλοποιούνται επιστρέφοντας το ζεύγος των κρυπτογραφημένων τιμών σε έναν τυχαίο Client. Ο τυχαίος Cient αποκρυπτογραφεί τις τιμές και εκτελεί τον πολλαπλασιασμό, στην συνέχεια κρυπτογραφεί το αποτέλεσμα και το επιστρέφει στον Server. Έπειτα ο Server συνεχίζει με τους υπολογισμούς των αποστάσεων μεταξύ των συναγερμών.

Στο τέλος αυτού του βήματος ο Server έχει υπολογίσει τις αποστάσεις μεταξύ όλων των ζευγών συναγερμών, ανεξάρτητα απο ποιον Client προέρχονται αυτοί οι συναγερμοί.

4.1.4.3 Ο Server εκτελεί ομαδοποίηση

Μόλις όλες οι αποστάσεις είναι διαθέσιμες στον Server, ο αλγόριθμος ομαδοποίησης μπορεί να εκτελεστεί. Όπως έχει περιγραφεί στο υποκεφάλαιο 3.1.1.2, ο αλγόριθμος k-medoids δημιουργεί Clusters των οποίων τα κέντρα επιλέγονται από τα σημεία (συναγερμούς) που πρέπει να ομαδοποιηθούν. Η απόσταση ενός σημείου (συναγερμού) από το κέντρο ενός Cluster υπολογίζεται από την απόσταση μεταξύ σημείων (συναγερμών) από το αρχικό σύνολο δεδομένων. Το προηγούμενο βήμα έχει δημιουργήσει όλες τις απαιτούμενες αποστάσεις, οπότε η επαναλαμβανόμενη διαδικασία διαμόρφωσης των Clusters δεν χρειάζεται να πραγματοποιήσει υπολογισμούς σχετικά με τις αποστάσεις.

Από την άλλη πλευρά κατά τη διάρκεια της ομαδοποίησης, αυτές οι αποστάσεις πρέπει να συγκριθούν μεταξύ τους. Κάθε σημείο (συναγερμός) αποδίδεται στο Cluster, το κέντρο του οποίου είναι το πλησιέστερο μεταξύ όλων των υποψήφιων Clusters. Αυτό σημαίνει ότι ο Server πρέπει να διεξάγει συγκρίσεις μεταξύ κρυπτογραφημένων τιμών. Στην πράξη, ο Server πρέπει να επιλέξει το ελάχιστο από μια σειρά κρυπτογραφημένων τιμών. Όταν συμβαίνει αυτό, ο Server ζητά από έναν τυχαίο Client να επιλέξει την ελάχιστη τιμή. Ο Client επιστρέφει μια κρυπτογραφημένη έκδοση ενός διανύσματος, τα στοιχεία του οποίου είναι όλα ίσα με το μηδέν εκτός από εκείνο το οποίο αντιστοιχεί στην ελάχιστη τιμή, και το οποίο είναι ίσο με την μονάδα.

Ο Server είναι ικανός να τρέξει όλους τους απαιτούμενους γύρους για τον αλγόριθμο k-medoids, και να παράξει μια τελική κατανομή των συναγερμών στους k διαφορετικούς Clusters.

4.1.4.4 Ο Server επιστρέφει τα αποτελέσματα

Έπειτα ο Server μπορεί να επιστρέψει στους Clients πληροφορίες σχετικά με συμβάντα ασφαλείας που έχουν συμβεί σε όλο το δίκτυο στο τελευταίο χρονικό διάστημα. Σύμφωνα με τις αρχικές ρυθμίσεις, μπορεί να αποκαλυφθεί διαφορετικός όγκος πληροφορίας. Για παράδειγμα ο Server μπορεί να φανερώσει σε έναν Client:

- μόνο τον όγκο των συναγερμών του Cluster στο οποίο ανήκει κάποιος συναγερμός του Client.
- τις IPs των συναγερμών σε ένα Cluster στο οποίο ανήκει κάποιος συναγερμός του Client.

- Όλα τα δεδομένα σχετικά με τους συναγερμούς σε ένα Cluster στο οποίο ανήκει κάποιος συναγερμός του Client.
- όλα τα δεδομένα σχετικά με τους συναγερμούς σε όλα τα Clusters

4.2 Υλοποίηση

4.2.1 Βασικές παράμετροι

Προκειμένου να αναλυθεί η υλοποίηση της προτεινόμενης μεθοδολογίας, οι βασικές παράμετροι που χρησιμοποιούνται παρουσιάζονται στην παρούσα υποενότητα. Υποθέτουμε ότι έχουμε m συναγερμούς και πως ο αλγόριθμος k-medoids θα δημιουργήσει k Clusters.

Ο **distances DIST (mxm)** είναι ένας τετραγωνικός πίνακας που περιέχει τις αποστάσεις μεταξύ όλων των m συναγερμών. Ο πίνακας DIST είναι συμμετρικός, καθώς η απόσταση μεταξύ δύο συναγερμών είναι η ίδια ανεξάρτητα από τη σειρά με την οποία λαμβάνονται υπόψη αυτοί οι δύο συναγερμοί στον υπολογισμό.

Ο **clusters' centers CC (mxk)** είναι ένας πίνακας που αποθηκεύει ποιος συναγερμός είναι το κέντρο κάθε Cluster. Κάθε στήλη του πίνακα αντιστοιχεί σε ένα Cluster. Κάθε στήλη περιέχει μόνο ένα στοιχείο ίσο με την μονάδα, το στοιχείο της γραμμής που αντιστοιχεί στο συναγερμό που είναι το κέντρο του Cluster. Όλα τα άλλα στοιχεία της στήλης είναι ίσα με μηδέν.

Ο **distance from clusters DFC (mxk)** είναι ένας πίνακας που αποθηκεύει την απόσταση κάθε συναγερμού από κάθε cluster. Κάθε γραμμή αντιστοιχεί σε κάθε συναγερμό. Τα στοιχεία της γραμμής είναι ίσα με τις αποστάσεις του αντίστοιχου συναγερμού από όλα τα Clusters (σύμφωνα με τη στήλη του στοιχείου).

Ο **belong to cluster BTC (mxk)** είναι ένας πίνακας που αποθηκεύει σε ποιο Cluster ανήκει κάθε συναγερμός. Κάθε γραμμή αντιστοιχεί σε κάθε έναν από τους συναγερμούς. Κάθε γραμμή περιέχει ένα στοιχείο ίσο με την μονάδα, το οποίο αντιστοιχεί στο Cluster που ανήκει ο συναγερμός, ενώ όλα τα άλλα στοιχεία της γραμμής είναι ίσα με το μηδέν.

Ο **clusters groups CG (mxm)** είναι ένας τετραγωνικός πίνακας που εμφανίζει ποιοι συναγερμοί βρίσκονται στο ίδιο Cluster. Κάθε σειρά αντιστοιχεί σε κάθε συναγερμό και κάθε στήλη αντιστοιχεί επίσης σε κάθε συναγερμό. Ένα στοιχείο σε αυτόν τον πίνακα είναι ίσο με τη μονάδα εάν οι συναγερμοί που αντιστοιχούν στη γραμμή και στη στήλη του

στοιχείου ανήκουν στο ίδιο Cluster. Διαφορετικά, η τιμή του στοιχείου είναι μηδέν.

Ο **distances in groups DG (mxm)** είναι ένας τετραγωνικός πίνακας που αποθηκεύει την απόσταση των συναγερμών από όλους τους συναγερμούς που ανήκουν στην ίδια ομάδα. Κάθε γραμμή αντιστοιχεί σε κάθε συναγερμό και κάθε στήλη αντιστοιχεί επίσης σε κάθε συναγερμό. Ένα στοιχείο σε αυτόν τον πίνακα είναι ίσο με την απόσταση των συναγερμών που αντιστοιχούν στη γραμμή και στη στήλη του στοιχείου, αν αυτοί οι δύο συναγερμοί ανήκουν στο ίδιο Cluster. Διαφορετικά, η τιμή των στοιχείων είναι μηδέν.

Το **center metrics CM (mx1)** είναι ένα μονοδιάστατο διάνυσμα. Κάθε στοιχείο ανήκει σε κάθε συναγερμό και είναι ίσο με το άθροισμα όλων των αποστάσεων του συγκεκριμένου συναγερμού από τους άλλους συναγερμούς που ανήκουν στο ίδιο Cluster. Στην πράξη, είναι ένα μέτρο που χρησιμοποιείται για την ενημέρωση των κέντρων των Clusters.

$$cm_i = \sum dist(a_i, a_j), \forall j : a_j \in c^i \tag{4.4}$$

όπου c^i είναι το Cluster στο οποίο ανήκει το a_i.

Ο **cluster center metric CCM (mxk)** είναι ένας πίνακας που κάθε στήλη του αντιστοιχεί σε κάθε ένα από τα Clusters. Κάθε στοιχείο αποθηκεύει την υπολογισμένη κεντρική μέτρηση για τον συναγερμό που αντιστοιχεί στη γραμμή του στοιχείου, σε σχέση με το συγκεκριμένο Cluster. Οι τιμές των στοιχείων που αντιστοιχούν σε συναγερμούς, που δεν ανήκουν στο cluster της συγκεκριμένης στήλης, είναι ίσες με το μηδέν.

4.2.2 Απόσταση μεταξύ συναγερμών

Το κύριο μέτρο που χρησιμοποιείται για τη διεξαγωγή της ομαδοποίησης είναι η απόσταση μεταξύ των συναγερμών. Όσο περισσότερο μοιάζουν δύο συναγερμοί τόσο μικρότερη είναι η απόσταση μεταξύ τους. Έτσι, παρόμοιοι συναγερμοί τελικά καταλήγουν στα ίδια Clusters.

Κάθε συναγερμός κατέχει διάφορα δεδομένα όπως το id του, η source IP, η destination IP, η source port και η destination port. Προκειμένου να υπολογιστούν οι αποστάσεις των συναγερμών, χρησιμοποιήθηκαν δύο πεδία.

Το πρώτο είναι η εξωτερική διεύθυνση IP κάθε συναγερμού. Κάθε Client εξάγει από τα πεδία IP του συναγερμού τη διεύθυνση IP που δεν ανήκει στο δίκτυο του προστατευμένου οργανισμού. Αυτό μπορεί να εμφανιστεί είτε σε πεδία source IP είτε σε πεδία destination

IP και υποδηλώνεται ως IP_{ext}.

Το δεύτερο είναι η υπογραφή του συναγερμού (ή id συναγερμού). Αυτή είναι ένας ακέραιος αριθμός που αντιστοιχεί στην επίθεση που ανιχνεύεται και εμφανίζεται ως $alert_{sig}$.

Η απόσταση μεταξύ των συναγερμών υπολογίζεται βάσει κάποιων επιμέρους αποστάσεων.

Η πρώτη επιμέρους απόσταση $dist^{sig}$ σχετίζεται με την υπογραφή των συναγερμών. Είναι μία δυαδική τιμή η οποία είναι ίση με το 2 αν οι δύο συναγερμοί μοιράζονται την ίδια υπογραφή και ίση με το μηδέν σε οποιαδήποτε άλλη περίπτωση.

$$dist^{sig}(a_i, a_j) = \begin{cases} 2 & \text{if } a^i_{sig} = a^j_{sig} \\ 0 & \textit{otherwise} \end{cases} \tag{4.5}$$

Ο δεύτερος συντελεστής είναι η απόσταση IP που υπολογίζεται σύμφωνα με τα τέσσερα octets της διεύθυνσης IP. Εάν η εξωτερική διεύθυνση IP του συναγερμού a^i είναι $IP^i_{ext} : x^i_4.x^i_3.x^i_2.x^i_1$ τότε η απόσταση των διευθύνσεων IP είναι:

$$dist^{ip}(a_i, a_j) = 2^3 * |x^i_4 - x^j_4| + 2^2 * |x^i_3 - x^j_3| + 2^1 * |x^i_2 - x^j_2| + 2^0 * |x^i_1 - x^j_1| \tag{4.6}$$

Ο τρίτος συντελεστής είναι η γεωγραφική απόσταση των σημείων που παράγονται από το IP_{ext} κάθε συναγερμού. Η απόσταση αυτή εμφανίζεται ως $dist^{geo}$ και ο υπολογισμός της είναι η Ευκλείδια απόσταση των δύο σημείων σύμφωνα με τις συντεταγμένες τους x και y.

$$dist^{geo}(a_i, a_j) = (x^i_{geo} - x^j_{geo})^2 + (y^i_{geo} - y^j_{geo})^2 \tag{4.7}$$

Τέλος η απόσταση μεταξύ των συναγερμών $dist(a_i, a_j)$ υπολογίζεται ως εξής:

$$dist(a_i, a_j) = dist(a_i, a_j)^{sig} + dist(a_i, a_j)^{ip} + dist(a_i, a_j)^{geo} \tag{4.8}$$

4.2.3 Υπολογισμοί

Κάθε Client διαθέτει ένα ζευγάρι δημοσίου και ιδιωτικού κλειδιού Paillier. Αυτό το ζευγάρι κλειδιών είναι κοινό μεταξύ όλων των Clients έτσι ώστε η όλη διαδικασία να καθίσταται εφικτή, ενώ δεν είναι προσβάσιμη από τον Server.

4.2.3.1 Διανύσματα alerts και πίνακες dist

Αρχικά, οι Clients πρέπει να υποβάλλουν στο Server όλα τα δεδομένα σχετικά με τους συναγερμούς τους καθώς και τις αποστάσεις μεταξύ των τελευταίων. Κάθε Client c_i που κρατά m_i συναγερμούς για το συγκεκριμένο χρονικό παράθυρο, υποβάλλει στο Server ένα διάνυσμα μήκους m_i με δεδομένα συναγερμών:

$$alerts_i = [ad_1, ...ad_{m_i}] \tag{4.9}$$

όπου κάθε ad_j είναι ένα σετ δεδομένων σχετικών με το συγκεκριμένο συναγερμό

$$ad_j = (a^i_{sig}, x^j_4, x^j_3, x^j_2, x^j_1, x^j_{geo}, y^j_{geo}) \tag{4.10}$$

Στην εξίσωση 4.10 η μεταβλητή a^i_{sig} αντιπροσωπεύει την υπογραφή του συναγερμού. Η υπογραφή του συναγερμού αντιπροσωπεύεται από ένα διάνυσμα ίσο σε μήκος με τις πιθανές τιμές υπογραφής. Έχει μηδενικά και μόνο το στοιχείο που αντιστοιχεί στην υπογραφή του συναγερμού είναι ίσο με τη μονάδα.

Οι μεταβλητές $x^j_4, x^j_3, x^j_2, x^j_1$ αντιπροσωπεύουν τους τέσσερις συντελεστές της εξωτερικής διεύθυνσης IP του συναγερμού και x^j_{geo}, y^j_{geo} είναι οι γεωγραφικές συντεταγμένες της εξωτερικής διεύθυνσης IP του συναγερμού.

Αυτές οι τιμές υφίστανται υπο-δειγματοληψία προκειμένου να διατηρηθούν οι προκύπτουσες αποστάσεις μικρότερες. Έτσι, κάθε μία από τις πραγματικές συντεταγμένες x^j_{geo}, y^j_{geo} αντικαθίσταται από το αποτέλεσμα της ακέραιας διαίρεσης μεταξύ της τιμής και του αριθμού 3. Έτσι, το εύρος τιμών γίνεται (-30,30) αντί για (-90,90) για τιμές γεωγραφικού πλάτους και (-60,60) αντί για (-180,180) για τιμές γεωγραφικού μήκους.

Αν και αυτή η υπο-δειγματοληψία μειώνει την ακρίβεια των υπολογιζόμενων αποστάσεων, δεν επηρεάζει την αποτελεσματικότητα της προσέγγισής. Η κύρια ομοιότητα που παρατηρείται συνήθως μεταξύ επιθέσεων στον κυβερνοχώρο είναι ότι από την ίδια ή γειτονικές χώρες. Αυτή η πληροφορία θα είναι ευδιάκριτη ακόμη και με λιγότερες πιθανές τιμές για το γεωγραφικό πλάτος και το μήκος.

Επιπλέον ο Client υπολογίζει τις αποστάσεις μεταξύ των m_i συναγερμών και χτίζει έναν

τετραγωνικό πίνακα μεγέθους m_ixm_i που ονομάζεται $dist_i$ και περιέχει τα αποτελέσματα.

$$dist_i = \begin{bmatrix} d_{11} & d_{12} & \dots & d_{1m_i} \\ d_{21} & d_{22} & \dots & d_{2m_i} \\ \vdots & \vdots & \ddots & \vdots \\ d_{m_i1} & d_{m_i2} & \dots & d_{m_im_i} \end{bmatrix} \tag{4.11}$$

Στην εξίσωση 4.11 κάθε στοιχείο d_{ij} αντιστοιχεί στην υπολογισμένη απόσταση για τους συναγερμούς a_i, a_j.

Οι Clients κρυπτογραφούν όλα τα σετ στοιχείων και όλες τις αποστάσεις με το δημόσιο κλειδί Pailier και κατασκευάζουν τις κρυπτογραφημένες εκδόσεις του διανύσματος $alerts_i$ και του πίνακα $dist_i$ που δηλώνονται ως $e(alert_i)$ και $e(dist_i)$. Οι κρυπτογραφημένες δομές δεδομένων αποστέλλονται στο Server.

4.2.3.2 Δημιουργώντας τον πίνακα DIST

Ο Server λαμβάνει ένα ζευγάρι $e(alert_i)$ και $e(dist_i)$ από κάθε Client. Όταν όλοι οι Clients έχουν στείλει τα απαραίτητα δεδομένα, ο Server μπορεί να προχωρήσει στην κατασκευή του πίνακα DIST, ο οποίος διατηρεί τις αποστάσεις μεταξύ όλων των συναγερμων όλων των Clients. Οι γραμμές και οι στήλες του πίνακα DIST αντιστοιχούν σε συναγερμούς από όλο το σύνολο. Η ταξινόμηση αυτών των συναγερμών πρέπει να είναι τέτοια που θα επιτρέπει να ομαδοποιούνται από τον Client που τους έχει παράγει. Με άλλα λόγια, οι συναγερμοί ενός συγκεκριμένου Client πρέπει να εμφανίζονται σε συνεχόμενο εύρος στην ευρετηρίαση των γραμμών και των στηλών του πίνακα DIST.

Μερικοί υπο-πίνακες του πίνακα DIST έχουν ήδη υπολογιστεί από τους Clients στους $dist_i$ πίνακες. Αυτοί οι πίνακες ευθυγραμμίζονται κατά μήκος της διαγωνίου του πίνακα DIST. Τα υπόλοιπα στοιχεία πρέπει να υπολογιστούν από το Server. Αυτοί οι υπολογισμοί διεξάγονται χρησιμοποιώντας τις κρυπτογραφημένες τιμές που έχουν αποθηκευτεί στα διανύσματα $alerts_i$.

Στην περίπτωση που οι συναγερμοί που πρέπει να συγκριθούν είναι οι a_i και a_j, ο Server έχει να κάνει τους ακόλουθους υπολογισμούς.

Ο υπολογισμός της $dist^{sig}$ διεξάγεται με πολλαπλασιασμό των δύο διανυσμάτων υπογραφής a^i_{sig},a^j_{sig} στοιχείο με στοιχείο και μετά αθροίζοντας όλα τα στοιχεία του διανύσματος που προκύπτει. Σε περίπτωση που οι δύο υπογραφές διανυσμάτων είναι πανομοιότυπες τότε το αποτέλεσμα της διαδικασίας είναι ίσο με 2, ενώ διαφορετικά είναι ίσο με 0.

Κατά την διάρκεια αυτής της διεργασίας ο Server χρειάζεται να κάνει μερικές προσθέσεις και πολλαπλασιασμούς μεταξύ κρυπτογραφημένων τιμών. Ο Server δύναται να εκτελέσει τις προσθέσεις, όπως έχει αναλυθεί στην υποενότητα 3.3, αλλά προκειμένου να εκτελέσει τους πολλαπλασιασμούς, πρέπει να τους αναθέσει στους Clients με τυχαίο τρόπο.

Όσον αφορά τον υπολογισμό της απόστασης $dist^{ip}(a_i, a_j)$, ο Server πρέπει να πραγματοποιήσει μερικές αφαιρέσεις, οι οποίες είναι εφικτές ακόμη και αν τα δεδομένα είναι κρυπτογραφημένα σύμφωνα με την υποενότητα 3.3. Επιπλέον πρέπει αποφασίσει για την απόλυτη τιμή των ακεραίων και στην περίπτωση αυτή θα πρέπει να ζητήσει από τους Clients να παρέμβουν.

Αν ο Server θέλει να υπολογίσει την απόλυτη τιμή $|z|$ του ακεραίου αριθμού z υπολογίζει και τους z και $-z$, προσθέτει και στους δύο έναν τυχαίο αριθμό $r, r > max(z)$. Αυτό είναι εφικτό καθώς ο Server μπορεί να προσθέσει έναν μη κρυπτογραφημένο αριθμό με έναν κρυπτογραφημένο, σύμφωνα με την υποενότητα 3.3. Κατόπιν στέλνει τόσο το $z + r$ και το $-z + r$, σε κρυπτογραφημένη μορφή, ο Client τα αποκρυπτογραφεί και επιστρέφει ως απάντηση ποιο από τα δύο είναι το μεγαλύτερο. Αν $z + r$ είναι μεγαλύτερο επιστρέφει $|z| = z$, αλλιώς $|z| = -z$.

Ο Client δεν γνωρίζει τις πραγματικές τιμές των αριθμών, ούτε την σειρά τους, οπότε δεν μπορεί να υποθέσει τίποτα για τον αριθμό, η απόλυτη τιμή του οποίου πρέπει να υπολογιστεί.

Τέλος, όσον αφορά την απόσταση των γεωγραφικών σημείων $dist^{geo}(a_i, a_j)$, ο Server πρέπει να κάνει αφαιρέσεις και μία προσθεση, οι οποίες είναι εφικτές, ενώ πρέπει να υπολογίσει τα τετράγωνα των δύο αριθμών, κάτι το οποίο πρέπει να γίνει από τους Clients. Και πάλι, για να προτατευτεί η πραγματική τιμή για την οποία πρέπει να υπολογιστεί το τετράγωνο, ο Server προσθέτει έναν τυχαίο αριθμό $r, r > 0$. Στην πράξη, εάν χρειάζεται να υπολογίσει το τετράγωνο του αριθμού z, στέλνει στον Client την τιμή $z + r$. Ο Client αποκρυπτογραφεί την τιμή, υπολογίζει το τετράγωνο, το κρυπτογραφεί και το στέλνει πίσω, αλλά δεν μπορεί να γνωρίζει την πραγματική τιμή του z. Ο Server λαμβάνει την τιμή $(z+r)^2$ και αφαιρεί από αυτό την τιμή του $r^2 + 2 * r * z$, την οποία μπορεί να υπολογίσει καθώς περιέχει έναν πολλαπλασιασμό μιας κρυπτογραφημένης τιμής με μία μη-κρυπτογραφημένη και την πρόσθεση μιας κρυπτογραφημένης τιμής με μια μη-κρυπτογραφημένη τιμή.

$$z^2 = (z + r)^2 - (r^2 + 2 * r * z) \tag{4.12}$$

Τέλος, η γενική απόσταση μεταξύ των συναγερμών υπολογίζεται ως ένα άθροισμα των τριών μερικών αποστάσεων που είναι κάτι που μπορεί να κάνει ο Server. Οι τρεις αποστάσεις υφίστανται κανονικοποίηση, για να έχουν την ίδια βαρύτητα στο τελικό αποτέλεσμα.

$$max(dist^{geo}) = 60^2 + 120^2 = 18000 \tag{4.13}$$

$$max(dist^{ip}) = 8 * 255 + 4 * 255 + 2 * 255 + 255 = 3825 \tag{4.14}$$

$$max(dist^{sig}) = 2 \tag{4.15}$$

Έτσι, η πραγματική απόσταση μεταξύ των δύο συναγερμών υπολογίζεται ως εξής:

$$dist^{norm}(a_i, a_j) = nf^{sig} * dist(a_i, a_j)^{sig} + nf^{ip} * dist(a_i, a_j)^{ip} + dist(a_i, a_j)^{geo} \tag{4.16}$$

όπου $nf^{sig} = 18000/2 = 9000$ and $nf^{ip} = 18000/3825 \simeq 5$

4.2.3.3 Ένας τυπικός γύρος k-medoids

Μετά το αρχικό στάδιο υπολογισμού όλων των τιμών για τον πίνακα DIST, ο Server μπορεί να συνεχίσει με την ουσιαστική διαδικασία ομαδοποίησης. Αυτή είναι μια επαναληπτική διαδικασία που εκτελεί τους απαιτούμενους γύρους k-medoids, μέχρι να οριστικοποιηθούν τα τελικά Clusters συναγερμών. Πριν ξεκινήσει αυτό, ο Server συμπληρώνει τον πίνακα CC (ο οποίος περιέχει τα κέντρα των Clusters) με τυχαίες τιμές.

Πίνακας DFC

Στην αρχή κάθε γύρου ο Server υπολογίζει τον πίνακα DFC, ο οποίος περιέχει την απόσταση κάθε σημείου από το κέντρο κάθε Cluster, πολλαπλασιάζοντας τους πίνακες CC και DIST. Ο πίνακας DIST είναι κρυπτογραφημένος, ενώ ο πίνακας CC όχι. Ο πολλαπλασιασμός των πινάκων απαιτεί πολλαπλασιασμούς μεταξύ μη-κρυπτογραφημένων και κρυπτογραφημένων τιμών, οι οποίοι είναι εφικτοί και προσθέσεις μεταξύ κρυπτογραφημένων τιμών, οι οποίες είναι επίσης εφικτές.

$$DFC = DIST * CC \tag{4.17}$$

Κάθε γραμμή του πίνακα DFC περιέχει k τιμές, οι οποίες είναι οι αποστάσεις του συναγερμού από κάθε ένα από τα n Clusters.

Πίνακας BTC

Το επόμενο βήμα είναι η κατασκευή του πίνακα BTC από τον πίνακα DFC. Για να συμβεί αυτό, ο Server πρέπει να βρει τη μέγιστη τιμή σε κάθε γραμμή του πίνακα BTC και να την αντικαταστήσει με μια τιμή μονάδας, ενώ όλα τα υπόλοιπα στοιχεία της γραμμής να είναι ίσα με το μηδέν. Αυτό απαιτεί πολλαπλές συγκρίσεις μεταξύ των τιμών της κάθε γραμμής, οι οποίες δεν μπορούν να πραγματοποιηθούν από το Server, καθώς αυτές οι τιμές είναι κρυπτογραφημένες.

Ο Server εξάγει τις γραμμές του πίνακα DFC ως διανύσματα και διεξάγει τυχαίες ανταλλαγές μεταξύ των τιμών τους. Οι ανταλλαγές αποθηκεύονται, προκειμένου να χρησιμοποιηθούν για την αποκατάσταση των δεδομένων στην αρχική τους μορφή. Ο Server προσθέτει επίσης μια τυχαία τιμή $r, r > 0$ σε όλες τις τιμές του διανύσματος, προκειμένου να τις αποκρύψει από τους Clients. Το προκύπτων διάνυσμα αποστέλλεται στον Client, ο οποίος αποκρυπτογραφεί τις τιμές και εντοπίζει τη θέση της μέγιστης τιμής. Στη συνέχεια επιστρέφει ένα διάνυσμα με κρυπτογραφημένες μηδενικές τιμές σε όλες τις θέσεις εκτός από εκείνη στην οποία βρέθηκε η μέγιστη τιμή, όπου αποθηκεύει την τιμή της μονάδας σε κρυπτογραφημένη μορφή. Ο Server λαμβάνει τότε το κρυπτογραφημένο διάνυσμα, επαναφέρει τα στοιχεία που είχαν ανταλλαχθεί και συνενώνει όλα τα διανύσματα στον κρυπτογραφημένο πίνακα BTC.

Πίνακας CG

Ο πίνακας CG είναι ένας τετραγωνικός πίνακας που εμφαίνει για κάθε συναγερμό a_i, ποιοι άλλοι συναγερμοί ανήκουν στην ίδια ομάδα με τον a_i. Αυτό μπορεί εύκολα να υπολογιστεί από τον πίνακα BTC πολλαπλασιάζοντάς τον με τον ανάστροφο του $BTC^{\intercal}$.

$$CG = BTC * BTC^{\intercal} \tag{4.18}$$

Αυτός ο υπολογισμός απαιτεί ο Server να πραγματοποιεί πολλαπλασιασμούς και προσθέσεις μεταξύ κρυπτογραφημένων τιμών. Για τους πολλαπλασιασμούς πρέπει να χρησιμοποιήσει τους Clients. Πριν από την αποστολή των τιμών x και y για πολλαπλασιασμό, προσθέτει και στις δύο τυχαίες τιμές $r_x, r_x > 0$ και $r_y, r_y > 0$.

Ο Client υπολογίζει τότε το γινόμενο $(x + r_x) * (y + r_y)$ και το επιστρέφει κρυπτογραφημένο στο Server, χωρίς να είναι σε θέση να γνωρίζει τις πραγματικές τιμές των x, y. Τέλος,

ο Server χρησιμοποιεί την επιστρεφόμενη τιμή για να υπολογίσει το γινόμενο $x*y$ ως εξής:

$$x * y = (x + r_x)(y + r_y) - (r_x * y + r_y * y + r_x * r_y) \tag{4.19}$$

Ο Server είναι σε θέση να υπολογίσει τόσο το γινόμενο όσο και το άθροισμα μιας μη-κρυπτογραφημένης τιμής με μια κρυπτογραφημένη τιμή, όπως αναλύθηκε στην υποενότητα 3.3.

Πίνακας DG

Για το επόμενο βήμα ο Server πρέπει να υπολογίσει τον πίνακα DG που είναι ίσος με το γινόμενο Hadamard [4, 12] των CG και DIST. Ο Server πολλαπλασιάζει τους πίνακες CG και DIST, στοιχείο επί στοιχείο.

$$DG = CG \odot DIST \tag{4.20}$$

Σε αυτό το βήμα ο Server πρέπει να αναθέσει στους Clients τον πολλαπλασιασμό μεταξύ των κρυπτογραφημένων τιμών σύμφωνα με τον μηχανισμό που περιγράφηκε παραπάνω.

Διάνυσμα CM

Το διάνυσμα CM διατηρεί μία τιμή για κάθε συναγερμό. Αντιστοιχεί στην μετρική που καθορίζει την καταλληλότητα του συναγερμού ως κέντρο του Cluster στο οποίο ανήκει. Στην πράξη για κάθε συναγερμό αυτή η μετρική είναι το άθροισμα των αποστάσεων από τους υπόλοιπους συναγερμούς που ανήκουν στο ίδιο cluster.

Το διάνυσμα CM υπολογίζεται αθροίζοντας τα στοιχεία κάθε γραμμής του πίνακα DG. Απαιτούνται μόνο προσθέσεις, οπότε ο Server μπορεί να υπολογίσει το διάνυσμα CM από μόνος του.

Πίνακας CCM

Προκειμένου να δημιουργηθεί ο πίνακας CCM, απαιτείται ένας διαγώνιος πίνακας μεγέθους mxm που περιέχει το διανύσματα CM κατά μήκος της διαγωνίου. Ο Server κατασκευάζει τον πίνακα και στη συνέχεια τον πολλαπλασιάζει με τον πίνακα BTC.

$$CCM = diag(CM) * BTC \tag{4.21}$$

Ο πίνακας diag(CM) μπορεί να κατασκευαστεί στοιχείο ανά στοιχείο από το Server.

Για τον πολλαπλασιασμό των πινάκων ο Server αναθέτει τους πολλαπλασιασμούς μεταξύ των κρυπτογραφημένων στοιχείων στους Clients.

Ο προκύπτων πίνακας CCM περιέχει μία στήλη για κάθε ένα από τα Clusters k. Κάθε τέτοια στήλη διατηρεί μετρήσεις για όλους τους συναγερμούς του Cluster, οι οποίες πρέπει να συγκριθούν και να επιλεχθεί εκείνη με την ελάχιστη τιμή. Στις θέσεις των συναγερμών που δεν ανήκουν στο Cluster η τιμή είναι μηδενική.

Πίνακας CC

Στο τελικό στάδιο πρέπει να υπολογίστει εκ νέου ο πίνακας CC που περιέχει το κέντρο για κάθε Cluster. Για να συμβεί αυτό, ο Server πρέπει να επιλέξει την ελάχιστη τιμή σε κάθε στήλη του πίνακα CCM. Αλλά αυτό πρέπει να συμβεί χωρίς να ληφθούν υπόψη οι μηδενικές τιμές που θα αντιστοιχούν σε συναγερμούς που δεν ανήκουν στο Cluster.

Ο Server προσθέτει μια τυχαία τιμή $r, r > 0$ σε όλα τα στοιχεία σε μια στήλη, ανταλλάσσει τα στοιχεία για να μπερδέψει την σειρά τους πριν τα στέλνει στον Client. Ο Client αποκρυπτογραφεί όλα τα στοιχεία και επιλέγει το ελάχιστο στοιχείο. Θέτει τη σχετική θέση ίση με τη μονάδα και τα υπόλοιπα στοιχεία ίσα με μηδέν. Επιστρέφει τη στήλη στο Server. Ο Server εκτελεί τελικά τις αντίστροφες ανταλλαγές. Στη συνέχεια συνενώνει όλες τις στήλες για να δημιουργήσει τον τελικό πίνακα CC, ο οποίος δεν είναι κρυπτογραφημένος.

Αυτός ο τυπικός γύρος k-medoids επαναλαμβάνεται, μέχρις ότου ο προκύπτων πίνακας CC να είναι ίδιος με εκείνον του προηγούμενου γύρου. Στη συνέχεια, ο αλγόριθμος ομαδοποίησης έχει συγκλίνει και ο Server μπορεί να ειδοποιήσει τους πελάτες για τα αποτελέσματα.

Σύμφωνα με το επίπεδο συνεργασίας μεταξύ των Clients, το ποσό των πληροφοριών που επιστρέφονται σε αυτούς μπορεί να διαφέρει. Ο Server περιέχει όλες τις κρυπτογραφημένες πληροφορίες σε μορφή που εμφαίνεται στην εξίσωση 4.10, μαζί με την κατανομή των συναγερμών σε Clusters. Έτσι, μπορεί να ειδοποιεί τους Clients μόνο για μεγάλα Clusters, μπορεί να στείλει μόνο το πλήθος των Clusters, την ταυτότητα των Clients των οποίων οι συναγερμοί αποτελούν ένα Cluster ή τα πραγματικά κρυπτογραφημένα δεδομένα των συναγερμών στο Cluster.

5. ΠΕΙΡΑΜΑΤΑ, ΥΛΟΠΟΙΗΣΗ ΚΑΙ ΑΠΟΤΕΛΕΣΜΑΤΑ

Έγινε μια προσπάθεια υλοποίησης του θεωρητικού μοντέλου που παρουσιάστηκε στο προηγούμενο κεφάλαιο με την χρήση της γλώσσας Python. Η επιλογή της συγκεκριμένης γλώσσας προγραμματισμού, έγινε με βάση την πληθώρα δωρεάν διαθέσιμων βιβλιοθηκών σχετικών με μαθηματικές πράξεις πινάκων, ομομορφική κρυπτογράφηση και επικοινωνία σε δικτυακό επίπεδο. Η χρήση αυτών των ήδη υλοποιημένων βιβλιοθηκών διευκόλυνε σε μεγάλο βαθμό την διαδικασία ανάπτυξης του λογισμικού, καθώς δεν χρειάστηκε η δημιουργία βασικών εργαλείων από την βάση τους. Ιδιαίτερη αναφορά πρέπει να γίνει στον υλοποιημένο κώδικα κρυπτογράφησης Paillier που χρησιμοποιήθηκε, τον οποίο έχει δημιουργήσει ο Mike Ivanov [13] και παρέχεται ελεύθερα στο GitHub. Υλοποιήθηκε το μεγαλύτερο μέρος του θεωρητικού μοντέλου με σκοπό την διερεύνηση της απόδοσης ενός τέτοιου συνεργατικού συστήματος σε πραγματικές συνθήκες. Η υλοποίηση είναι διαθέσιμη στο ακόλουθο repository στο GitHub: **https://github.com/gtheodoridis/pp-colab-ids**

5.1 Data-set

Προκειμένου να δοκιμαστεί η προτεινόμενη υλοποίηση, απαιτήθηκαν ρεαλιστικά δεδομένα κίνησης δικτύου. Το σύνολο δεδομένων που χρησιμοποιήθηκε ήταν το data-set UNB ISCX 2012 [20] που είναι σχετικό με ανίχνευση εισβολών. Πρόκειται για μια πρόσφατη προσέγγιση βασισμένη στην δημιουργία προφίλ τα οποία περιέχουν λεπτομερείς περιγραφές των εισβολών και αφηρημένων μοντέλων σχετικών με εφαρμογές, πρωτόκολλα ή οντότητες δικτύου χαμηλότερου επιπέδου.

Το data-set είναι πιο ρεαλιστικό από τα περισσότερα data-set που έχουν χρησιμοποιηθεί στο παρελθόν. Πρόκειται για ένα δίκτυο ενός οργανισμού με πέντε διαφορετικά εσωτερικά δίκτυα και μία ζώνη DMZ. Η ζώνη DMZ είναι μία ενδιάμεση ζώνη μεταξύ ενός δικτύου και του internet. Εκεί εγκαθίσταται οι servers που πρέπει να είναι προσβάσιμοι από το internet, οπότε στο εσωτερικό δίκτυο μπορεί να απαγορεύεται οποιαδήποτε πρόσβαση, με αποτέλεσμα μεγαλύτερη ασφάλεια.

Η προσέγγιση που χρησιμοποιείται για τη δοκιμή της ομαδοποίησης στο δίκτυο οργανισμών είναι να υποθέσουμε ότι κάθε ένα από τα έξι διαφορετικά υποδίκτυα ανήκει σε διαφορετικό οργανισμό και ότι η προστασία της ιδιωτικότητας της κίνησης δικτύου πρέπει να προστατεύεται κατά τη διάρκεια της ομαδοποίησης των συναγερμών.

Η κίνηση πακέτων διαχωρίστηκε σύμφωνα με το εύρος των IP των έξι διαφορετικών δικτύων, σε έξι διαφορετικά κομμάτια κυκλοφορίας. Για κάθε ένα από τα διαφορετικά δίκτυα χρησιμοποιείται ένας συνδυασμός ενός αισθητήρα Snort και ενός Client του προτεινόμενου συστήματος. Κάθε ένα από τα έξι κομμάτια κίνησης έχει χρησιμοποιηθεί ως είσοδος για τον αντίστοιχο αισθητήρα Snort. Με τον τρόπο αυτό προσομοιώνεται μια συνεργασία μεταξύ έξι διαφορετικών οργανισμών προκειμένου να δοκιμαστεί η αποτελεσματικότητα του προτεινόμενου συστήματος. Επιπλέον, η συσχέτιση μεταξύ της κίνησης των διαφορετικών δικτύων είναι σχετικά υψηλή, πράγμα που δημιουργεί μια κατάλληλη πλατφόρμα δοκιμών για την ομαδοποίηση.

5.2 Ανάλυση υλοποίησης

5.2.1 Client

Ανοίγουμε το αρχείο του data-set και διαβάζουμε τα δεδομένα, διαχωρίζουμε τα επιμέρους στοιχεία και τα φέρνουμε στην μορφή που τα χρειαζόμαστε για να τα χρησιμοποιήσουμε.

Ανοίγουμε ένα αρχείο που έχουμε δημιουργήσει με τα κλειδιά που έχουμε παράξει τρέχοντας εξωτερικά τις συναρτήσεις του Mike Ivanov για παραγωγή κλειδιών Paillier. Το κάνουμε αυτό για να έχουμε σταθερά κλειδιά κρυπτογράφησης και αποκρυπτογράφησης. Μέσα από αυτήν την διαδικασία δημιουργούμε τα αντικείμενα **private key** και **public key**.

Συνεχίζοντας την εκτέλεση του προγράμματος, δημιουργούμε το αντικείμενο **operations** το οποίο παίρνει τα public και private keys και περιλαμβάνει τις ακόλουθες συναρτήσεις:

- Συνάρτηση **multi**, η οποία παίρνει ως όρισμα δύο κρυπτογραφημένους αριθμούς (a,b), τους αποκρυπτογραφεί, τους πολλαπλασιάζει, κρυπτογραφεί το αποτέλεσμα του πολλαπλασιασμού και το επιστρέφει.

- Συνάρτηση **multi1**, η οποία παίρνει ως όρισμα δύο αριθμούς (a,b) όπου μόνο ο αριθμός «a» είναι κρυπτογραφημένος. Αποκρυπτογραφεί το a, κάνει τον πολλαπλασιασμό με το b, κρυπτογραφεί το αποτέλεσμα και το επιστρέφει.

- Συνάρτηση **compare**, η οποία δέχεται ως όρισμα δύο αριθμούς (a,b), τους αποκρυπτογραφεί, τους συγκρίνει για να βρει τον μεγαλύτερο. Τους κρυπτογραφεί και επιστρέφει ένα διάνυσμα όπου στην πρώτη θέση βρίσκεται ο μεγαλύτερος από τους

δύο προηγούμενους αριθμούς. Στην περίπτωση που οι δύο αριθμοί είναι ίσοι, στην πρώτη θέση μπαίνει ο αριθμός **b**.

- Συνάρτηση **isGreater**, η οποία παίρνει δύο αριθμούς (a,b), τους συγκρίνει και γυρίζει «true» άμα ο «a» είναι μεγαλύτερος ή ίσος του «b».
- Συνάρτηση **findmax**, η οποία δέχεται ως όρισμα δύο αριθμούς (a,b), συγκρίνει τον αριθμό **a** με το **-100**, άμα είναι διαφορετικά το αποκρυπτογραφεί, το συγκρίνει με το **b** και γυρίζει κρυπτογραφημένο τον μεγαλύτερο αριθμό από τους δυο.
- Συνάρτηση **minussious**, η οποία παίρνει ως όρισμα τους αριθμούς (a,b), τους αποκρυπτογραφεί, κάνει την αφαίρεση **a-b** και επιστρέφει το αποτέλεσμα της πράξης κρυπτογραφημένο.
- Συνάρτηση **isEqual**, η οποία δέχεται ως όρισμα δύο κρυπτογραφημένους αριθμούς (a,b), τους αποκρυπτογραφεί, τους συγκρίνει και γυρίζει «true» ή «false» αν είναι ίσοι οι δύο αριθμοί ή όχι.

Στην συνέχεια μέσω της συνάρτησης **daemon()** της βιβλιοθήκης **Pyro**, κάνουμε register το αντικείμενο **operations** για να μπορεί να καλέσει με την χρήση ενός **URI** απομακρυσμένα ο Server τις συναρτήσεις του αντικειμένου.

Ξεκινάει την λειτουργία του ένας Client, στην αρχή για να δημιουργηθεί σύνδεση με τον Server πρέπει να εισάγουμε το μοναδικό **URI** του Server. Το σύστημα μας για να ξεκινήσει να τρέχει θα πρέπει να συνδεθούν τόσοι Clients όσοι έχουμε ορίσει εμείς στο Configuration. Μόλις συνδεθεί και ο τελευταίος Client η ροή του προγράμματος ξεκινάει με κάθε Client να τρέχει ταυτόχρονα μέσω **threading** και να εκτελεί τις ακόλουθες διεργασίες.

- Αρχικά παίρνει τους **N** συναγερμούς από το data-set του, τους κρυπτογραφεί με τις συναρτήσεις κρυπτογράφησης του Paillier και τους εισάγει σε μία λίστα που ονομάζουμε **D1**, στην ουσία η **D1** είναι ένα **numpy.array** που περιέχει τις κρυπτογραφημένες αποστάσεις των μεταξύ των συναγερμών. Αυτός ο υπολογισμός γίνεται μέσω της συνάρτησης **calculate_dist** η οποία παίρνει ως όρισμα τους **N** συναγερμούς και υπολογίζει αποστάσεις **Manhattan**. Δηλαδή την απόλυτη τιμή της διαφοράς τους, μέσω της συνάρτησης **pairwise_distance** της εξωτερικής βιβλιοθήκης **sklearn**.
- Επιστρέφοντας στον Client, δημιουργούμε το αντικείμενο **thread** το οποίο μας βοηθάει να συνδεόμαστε στον Server και παράλληλα να ακούμε για αιτήματα χρησιμοποίησης των συναρτήσεων του Client **operations** από τον Server.

Thread: είναι δύο συναρτήσεις που καλούνται ταυτόχρονα.

- Η **pyro_Run** συνδέεται με τον Server και καλεί την **travel_data** συνάρτηση του στέλνοντας του:

  1. μια λίστα με κρυπτογραφημένες ip
  2. τον np.array με τις αποστάσεις των συναγερμών κρυπτογραφημένες
  3. το **URI** του Client ώστε να το χρησιμοποιήσει ο Server στην περίπτωση που θέλει να καλέσει κάποια από τις συναρτήσεις του Client.

- Η **daemon_loop** περιμένει να ακούσει τον Server.

5.2.2 Server

Ο Server ξεκινάει την λειτουργία του δημιουργώντας τα δύο κλειδιά **private key** και **public key** χρησιμοποιώντας το αρχείο με τα κλειδιά του Paillier που έχουμε δημιουργήσει από πριν για διευκόλυνση της ροής του προγράμματος.

Έπειτα ο Server δημιουργεί το αντικείμενο **OneClassToRuleThemAll** κι στη συνέχεια εκτελεί παράλληλα δύο συναρτήσεις του αντικειμένου. Την **Run** και την **daemon**.

Η Run καλεί την συνάρτηση **calculation** η οποία περιμένει μέχρι να συνδεθούν όσοι Clients έχουμε ορίσει στο Configuration.

Η κλάση αυτή περιέχει τρεις μεταβλητές:

1. την **ArrayOfIPs** που περιέχει λίστες με ip
2. την **ArrayOfDistances** που είναι μία λίστα με πίνακες αποστάσεων
3. την **UriTable** που είναι μία λίστα με τα Uri των Clients.

Αφού συνδεθούν οι Clients , ο Server καλεί την συνάρτηση **dis** η οποία για κάθε client υπολογίζει τις αποστάσεις των ip του με αυτές των άλλων clients καλώντας την συνάρτηση **calc_dist**.

calc_dist: η συνάρτηση αυτή δέχεται ως όρισμα δύο λίστες-διανύσματα με ip. Την **va** με τις ip ενός Client και την **vb** με τις ip ενός άλλου Client. Στην συνέχεια προσπαθεί να υπολογίσει τις αποστάσεις όλων των στοιχείων μεταξύ τους. Για να γίνει αυτό επιλέγεται

τυχαία ένας Client και χρησιμοποιείται για να υλοποιήσει τις συναρτήσεις **compare** και **minussious** τις οποίες δεν μπορεί να υλοποιήσει ο Server γιατί έχει κρυπτογραφημένα δεδομένα.

Αφότου ο Client επιστρέψει τα αποτελέσματα των συναρτήσεων, ο Server είναι έτοιμος για να κάνει πρόσθεση κατά paillier με κρυπτογραφημένα δεδομένα και να δημιουργήσει έναν πίνακα **K** με τις κρυπτογραφημένες αποστάσεις των δύο Clients. Η συνάρτηση **calc_dist** επιστρέφει αυτόν τον πίνακα **K** στην **dis** η οποία τον τοποθετεί μέσα σε έναν μεγαλύτερο πίνακα **ArrayOfDistances** και τερματίζει την λειτουργία της.

Έπειτα ο Server καλεί την συνάρτηση **constract_final** η οποία παίρνει ως ορίσματα τους πίνακες **ArrayOfDistances** και **ArrayOfIPs** και δημιουργεί τον άνω τριγωνικό και τον κάτω τριγωνικο πίνακα χρησιμοποιώντας την μέθοδο hstack. Στο τέλος ενώνει τους τριγωνικούς και έχουμε τον τελικό πίνακα που ονομάζεται **final**.

Μετά από αυτήν την περίπλοκη αλγοριθμική διεργασία η **constract_final** επιστρέφει τον πίνακα **final** στην συνάρτηση **calculation** η οποία τον εκχωρεί στην μεταβλητή **D** και καλεί την συνάρτηση **clustering** με όρισμα τον **D**.

clustering: αυτή η συνάρτηση είναι αρμόδια για όλες τις διαδικασίες ομαδοποίησης του συστήματος. Αρχικά ορίζει τον αριθμό των clusters του συστήματος, και έπειτα δημιουργεί τον πίνακα **C** ($alerts_x clusters$) καλώντας την **draw_centers** ή οποία ορίζει τυχαία έναν συναγερμό ως κέντρο για κάθε cluster. Στη συνέχεια η **clustering** καλεί την συνάρτηση **center_positions** δίνοντας της σαν όρισμα τον πίνακα **C** με τα κέντρα.

Η **center_positions** με την σειρά της επιστρέφει μία λίστα **center_position_lst** με τις θέσεις που βρίσκονται τα κέντρα.

[(θέση κέντρου,cluster), (θέση κέντρου, cluster), (θέση κέντρου, ...)...].

Έπειτα ήρθε η στιγμή για την **clustering** να δημιουργήσει τον πίνακα **DFC** σε συντομογραφία του ”distance from cluster”.

Ο πίνακας αυτός δημιουργείται καλώντας την **arraymulti1** η οποία:

- Παίρνει ως ορίσματα:

  1. τον πίνακα **D** που περιέχει τις κρυπτογραφημένες αποστάσεις όλων των συναγερμών μεταξύ τους
  2. τον πίνακα **C** ο οποίος περιέχει **0** και **1** για τα κέντρα των clusters

- Ορίζει μηδενικό πίνακα **O** με διαστάσεις των αριθμό των γραμμών του **D** και των στηλών του **C**
- Πολλαπλασιάζει τα στοιχεία τους επιμέρους με την επιλογή ενός τυχαίου Client και χρήση της συνάρτησης **multi** του τελευταίου
- και επιστρέφει το αποτέλεσμα στον πίνακα **DFC**

Σειρά έχει ο πίνακας που δείχνει σε πιο cluster ανήκει ο κάθε συναγερμός, την δημιουργία αυτού του πίνακα που ονομάζουμε **BTC** αναλαμβάνει η συνάρτηση **calc_btc**.

Η **calc_btc**:

- Παίρνει σαν ορίσματα τον πίνακα **DFC** και έναν Client.
- Για κάθε γραμμή του **DFC** ελέγχει αν στη γραμμή υπάρχει κάποιο κέντρο.
  - Αν υπάρχει ορίζει ένα “low_value_index”.
  - Αλλιώς καλεί την συνάρτηση του Client **find_min** η οποία:
    * Δέχεται ως ορίσματα την γραμμή του **DFC** και έναν τυχαίο Client.
    * Καλεί την **compare** του Client, η οποία βρίσκει το ελάχιστο στοιχείο της γραμμής και το επιστρέφει σαν **k**.
  - Καλεί την συνάρτηση **min_pos** η οποία:
    * Δέχεται ως ορίσματα την γραμμή του **DFC**, το **k** και τον Client
    * Καλεί την **IsEqual** του Client επαναληπτικά μέχρι να βρει σε ποια θέση βρίσκεται η ελάχιστη τιμή-απόσταση της εκάστοτε γραμμής του **DFC** και επιστρέφει την θέση της ελάχιστης τιμής.
  - Αφού λοιπόν έχουμε εντοπίσει που βρίσκεται η ελάχιστη τιμή της γραμμής ή αν είναι κέντρο, μετατρέπουμε όλα τα στοιχεία της γραμμής σε μηδενικά εκτός από το στοιχείο της ελάχιστης τιμής (**low_value_index**). Έτσι δημιουργείται ο πίνακας **BTC** και επιστρέφεται στην συνάρτηση **clustering**.

Έχοντας πια όλη αυτή την πληροφορία, το πρόγραμμα μας αρχίζει να τρέχει μία συνθήκη επανάληψης κάνοντας τις ακόλουθες διεργασίες.

Αρχή Επανάληψης

- Υπολογισμός πίνακα **center_groups** πολλαπλασιάζοντας τον **BTC** με τον ανάστροφο **BTC**

$$BTC * BTC^{\intercal}$$

- Κλήση συνάρτησης **centerClusterDistance**, η οποία παίρνει ορίσματα τον πίνακα **centerGroups** και τον πίνακα **D**. Δημιουργεί μηδενικό πίνακα **M** διαστάσεων ανάλογων του **D**, πολλαπλασιάζει τον **D** με τον **centerGroups** χρησιμοποιώντας την **multi1** και επιστρέφει τον **M** που αποτελείται είτε από μηδενικά είτε από τιμές αποστάσεων. Σε αυτή την περίπτωση τα μηδενικά είναι μη-κρυπτογραφημένα ενώ οι αποστάσεις κρυπτογραφημένες.

- Δημιουργία μηδενικού πίνακα **met** διαστάσεων $1_x alerts$

- Αρχή εμφωλευμένης επανάληψης:

  - Η επανάληψη τρέχει για κάθε γραμμή του πίνακα **CGD** και αυτό που κάνει είναι να προσθέτει τις στήλες του.
  - Στο τέλος έχουμε τον πίνακα **met** που είναι στην ουσία μία στήλη που σε κάθε γραμμή έχει το άθροισμα όλων των στηλών της αντίστοιχης γραμμής του **CGD**. (τα μηδενικά μη-κρυπτογραφημένα και τους αριθμούς κρυπτογραφημένους)

- Τοποθέτηση του πίνακα **met** $1_x alerts$ ως διαγώνιο σε έναν μηδενικό πίνακα **diagMet** $alerts_x alerts$

- Πολλαπλασιασμός του πίνακα **diagMet** με τον πίνακα **BTC** και δημιουργία του πίνακα **firstComponentFinalForm** ($alerts_x clusters$). Ο πίνακας αυτός αποτελείται από μη-κρυπτογραφημένα μηδενικά και τις κρυπτογραφημένες αποστάσεις των alerts από τα clusters.

- Δημιουργία του πίνακα **preFinal** μέσω τις πρόσθεσης του **BTC** με έναν πίνακα ίδιων διαστάσεων αλλά γεμάτο άσσους και έπειτα τον πολλαπλασιασμό του αποτελέσματος με το **-10** έτσι ώστε να έχουμε **-10** εκεί που είχαμε άσσους και **0** εκεί που είχαμε μηδενικά.

- Δημιουργία πίνακα **finalForm**, προσθέτοντας τον **firstComponentFinalForm** και τον **prefinal** και έχουμε πίνακα διαστάσεων $alerts_x clusters$ ώπου αποτελείται από κρυπτογραφημένες αποστάσεις και μη-κρυπτογραφημένους αριθμούς: **-10**.

- Κλήση της **find_centers** η οποία παίρνει ως ορίσματα (alerts, clusters, finalform, met, public key) και βρίσκει την ελάχιστη τιμή της κάθε στήλης του **finalForm** και σε εκείνη την θέση τοποθετεί το καινούριο κέντρο του cluster.
- Αλλαγή της λίστας με τα κέντρα και επανυπολογισμός με την ίδια διαδικασία των πινάκων **DFC** και **BTC**.
- Έλεγχος του παλιού **BTC** αν είναι ίδιος με τον καινούριο.

Η διαδικασία της επανάληψης ολοκληρώνεται όταν ο πίνακας **BTC** είναι ίδιος με αυτόν που είχε προκύψει στην προηγούμενη επανάληψη.

5.2.3 Σενάρια εκτέλεσης αλγορίθμου

Όλη η διαδικάσια που αναλύθηκε σε βάθος παραπάνω χρησιμοποιήθηκε με σκοπό την εκτέλεση συγκεκριμένων σεναρίων χρήσης. Τα σενάρια αυτά διαμορφώθηκαν βάσει των εξής δύο παραγόντων:

- Τον αριθμό των Οργανισμών-Clients (Nodes) που συμμετέχουν στο συνεργατικό δίκτυο ανίχνευσης και ανάλυσης εισβολών
- Τον αριθμό των συναγερμών (alerts) που παρέχουν οι Clients για ομαδοποίηση.

Ειδικότερα, ο υλοποιημένος αλγόριθμος εκτελέστηκε για τρεις, τέσσερις, πέντε και έξι διαφορετικούς οργανισμούς. Κάθε μία εκτέλεση πλαισιώθηκε με έξι διαφορετικές υποεκτελέσεις με χρήση data-set που περιείχαν 10, 30, 60, 90 και 120 συναγερμούς για κάθε οργανισμό.

Όσον αφορά το hardware χρησιμοποιήθηκε ένας υπολογιστής με επεξεργαστή δύο πυρήνων χρονισμένο στα 2.4Ghz και 4GB RAM. Προφανώς η εκτέλεση των υλοποιημένων αλγορίθμων μπορεί να γίνει ταχύτερα έαν χρησιμοποιηθούν περισσότεροι υπολογιστικοί πόροι.

5.2.4 Αποτελέσματα

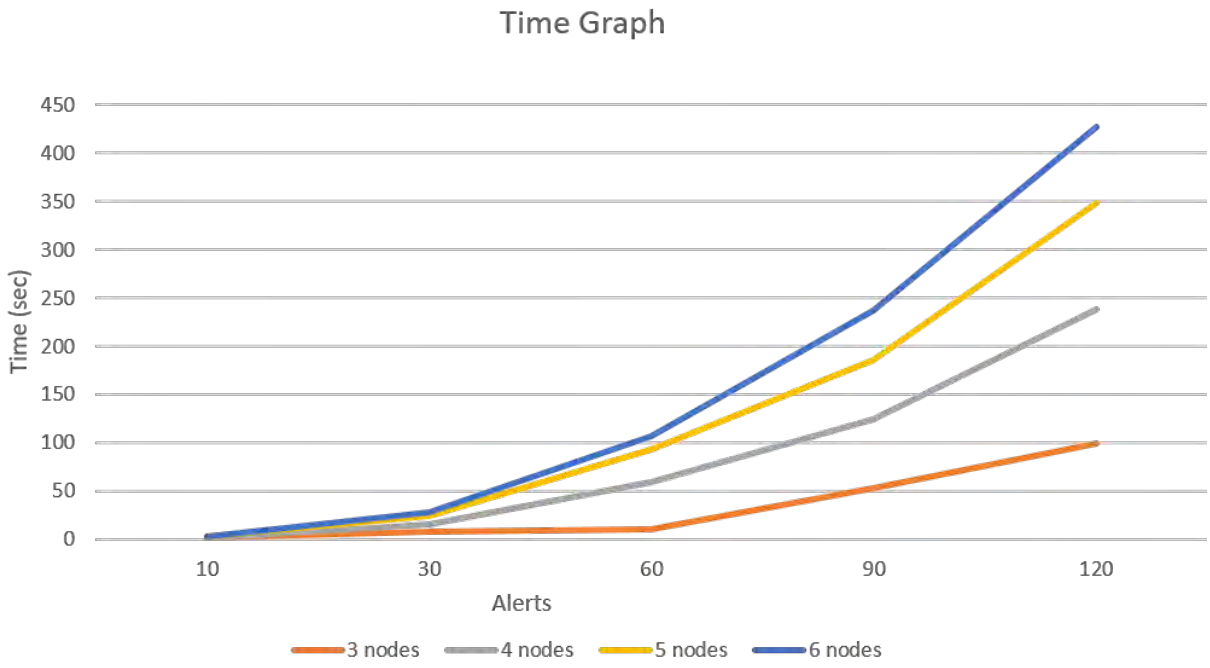
Με βάση τα αποτελέσματα των παραπάνω σεναρίων εκτέλεσης του αλγορίθμου, είμαστε σε θέση να εξάγουμε χρήσιμα συμπεράσματα για την απόδοση του συστήματος.

Ως κύριο μέτρο υπολογισμού της απόδοσης χρησιμοποιήθηκε ο χρόνος ολοκλήρωσης ενός γύρου ομαδοποίησης 5.1. Περαιτέρω μετρικά στοιχεία μας έδωσαν κατατοπιστικά στατιστικά για το πλήθος των πολλαπλασιασμών 5.2, των συγκρίσεων 5.3 και των αφαιρέσεων 5.4 που συνολικά εκτελέστηκαν. Τα στοιχεία αυτά σχετίζονται με τις στιγμές όπου απαιτείται οι εκάστοτε οργανισμοί να παρέχουν βοήθεια στην κεντρική μας οντότητα για την εκτέλεση των πράξεων που δεν μπορούν να εκτελεστούν ομομορφικά από την ίδια.

Στα αποτελέσματα αυτά όπως γίνεται αντιληπτό και παρακάτω στους πίνακες και τα γραφήματα είναι εμφανής η γραμμική αύξηση του χρόνου και των πράξεων όσο αυξάνονται οι Clients και οι συναγερμοί προς ομαδοποίηση.

Alerts Number	3 nodes	4 nodes	5 nodes	6 nodes
10	1,03	1,88	2,99	3,3
30	8,36	15,63	23,89	27,73
60	10	59,08	93,3	106,81
90	53,5	124,7	185	236,7
120	98,7	238,4	348	427,58

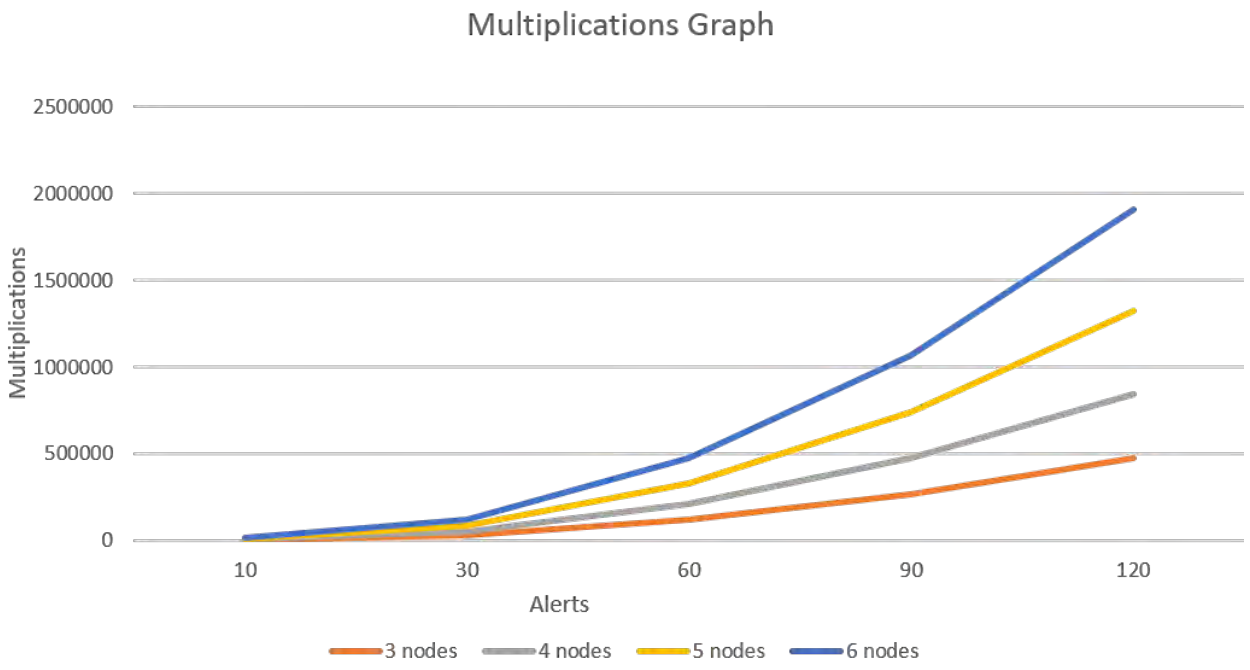
Πίνακας 5.1: Χρόνοι εκτέλεσης



Σχήμα 5.1: Γράφημα Χρόνου εκτέλεσης

Alerts Number	3 nodes	4 nodes	5 nodes	6 nodes
10	3300	5867	9167	13200
30	29700	52800	82500	118800
60	118800	211200	330000	475200
90	267000	475200	742500	1069200
120	475200	844800	1320000	1906800

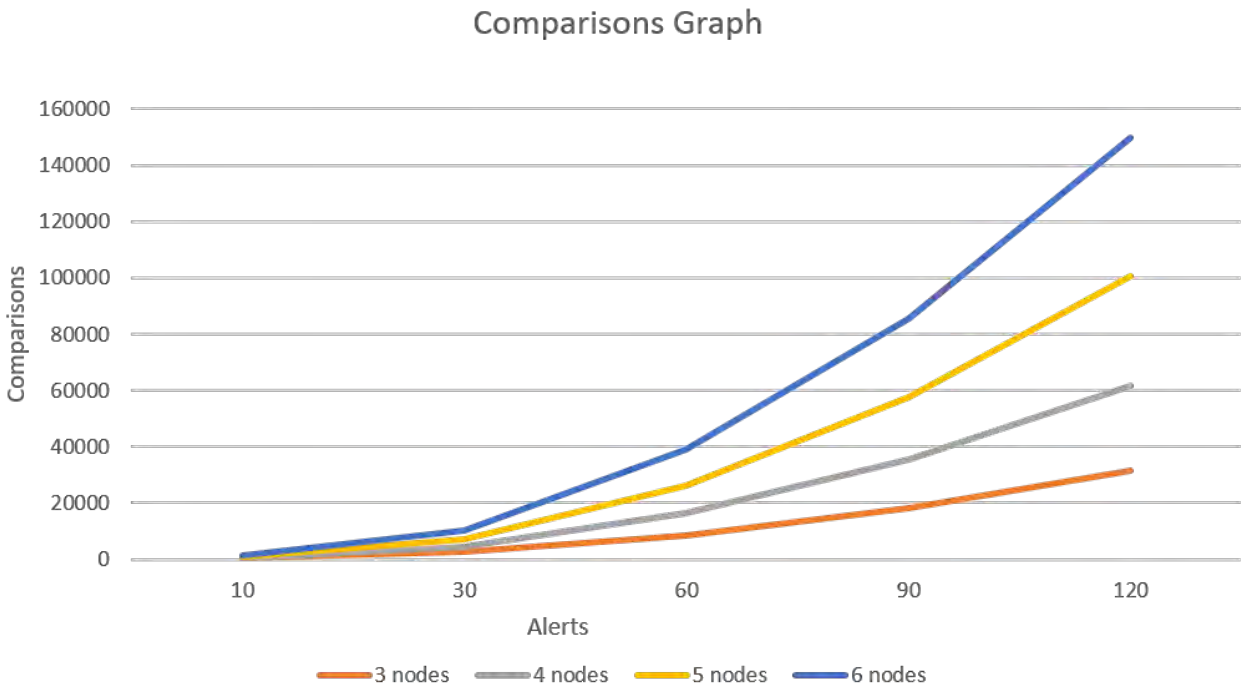
Πίνακας 5.2: Πλήθος πολλαπλασιασμών που εκτελέστηκαν



Σχήμα 5.2: Γράφημα πολ/σμών που εκτελέστηκαν

Alerts Number	3 nodes	4 nodes	5 nodes	6 nodes
10	428	714	1048	1474
30	2537	4556	7192	10450
60	8633	16270	26344	38901
90	18385	35284	57638	85330
120	31677	61548	100704	149692

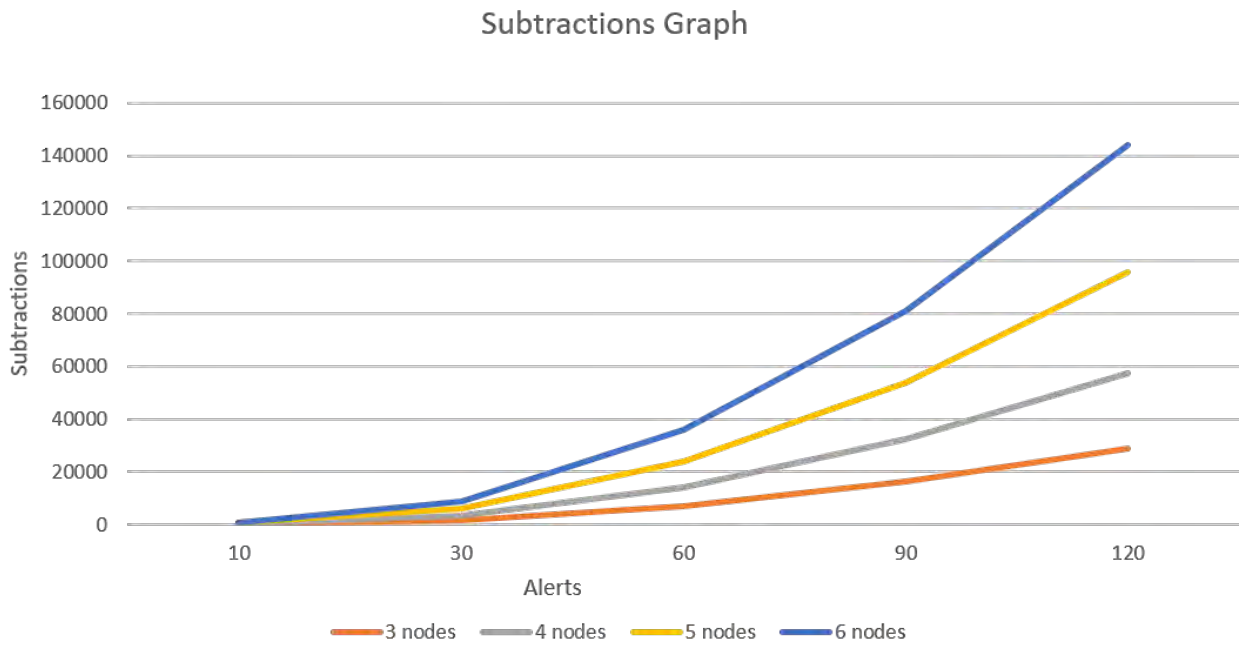
Πίνακας 5.3: Πλήθος συγκρίσεων που εκτελέστηκαν



Σχήμα 5.3: Γράφημα συγκρίσεων που εκτελέστηκαν

Alerts Number	3 nodes	4 nodes	5 nodes	6 nodes
10	200	400	667	1000
30	1800	3600	6000	9000
60	7200	14400	24000	36000
90	16200	32400	54000	81000
120	28800	57600	96000	144000

Πίνακας 5.4: Πλήθος αφαιρέσεων που εκτελέστηκαν



Σχήμα 5.4: Γράφημα αφαιρέσεων που εκτελέστηκαν

6. ΣΥΜΠΕΡΑΣΜΑΤΑ

Συμπερασματικά η υλοποίηση του αλγορίθμου, ικανοποίησε σε μεγάλο βαθμό τον αρχικό σκοπό της έρευνας και μελέτης πάνω στην διατήρηση της ιδιωτικότητας κατά την ομαδοποίηση συναγερμών συστημάτων ανίχνευσης εισβολών σε δίκτυα πολλαπλών οργανισμών. Η συνεισφορά της εργασίας σε θεωρητικό και πρακτικό επίπεδο κρίνεται σημαντική και ανοίγει τον δρόμο για ανάπτυξη και εξέλιξη αυτών των συστημάτων, πάντα με γνώμονα την ιδιωτικότητα των επιμέρους χρηστών.

Το συνεργατικό σύστημα ομαδοποίησης εισβολών που δημιουργήθηκε δείχνει να ανταποκρίνεται στις προσδοκίες μας για διατήρηση της ιδιωτικότητας των συνεργαζόμενων οργανισμών.

Η έμπιστη τρίτη οντότητα (Server) είναι ικανή να πραγματοποιεί την ομαδοποίηση κατά K-medoids εκτελώντας το μεγαλύτερο ποσοστό των απαραίτητων πράξεων και μετασχηματισμών από μόνη της. Το μοντέλο χρησιμοποίησης των οργανισμών για τις υπόλοιπες πράξεις που δεν καθίστανται ομομορφικά δυνατές από τον Server έχει μικρό κίνδυνο για απώλειες στην ιδιωτικότητα.

Η μεθοδολογία που ακολουθήσαμε αν και χρήζει περαιτέρω βελτίωσης έχει γερά μαθηματικά θεμέλια και διευκόλυνε αρκετά την υλοποίηση του αλγορίθμου και την μετέπειτα εκτέλεση των βασικών σεναρίων χρήσης.

Η απόδοση του συστήματος είναι εντυπωσιακή καθώς η διαδικασία της ομομορφικής κρυπτογράφησης και της χρήσης των ιδιοτήτων της είναι αρκετά κοστοβόρα σε υπολογιστική ισχύ. Παρόλα αυτά και λαμβάνντας υπόψιν το hardware που χρησιμοποιήθηκε για την εκτέλεση των πειραμάτων, οι χρόνοι εκτέλεσης είναι αρκετά ικανοποιητικοί και η αύξηση αυτών γραμμικά σε σχέση με την αύξηση του όγκου δεδομένων και οργανισμών στο δίκτυο μας καταδεικνύει ότι η προτεινόμενη προσέγγιση είναι εφικτό να χρησιμοποοηθεί και για μεγαλύτερους όγκους δεδομένων.

Σίγουρα η χρησιμοποίηση ενός απλού φορητού υπολογιστή μας περιόρισε αρκετά και αδυνάτισε σε έναν μικρό βαθμό την έρευνα μας.

6.1 Μελλοντική έρευνα

Όσον αφορά την μελλοντική έρευνα και ανάπτυξη πάνω στο ήδη υλοποιημένο μοντέλο, αυτή παρουσιάζει αρκετά ενδιαφέροντα σημεία.

Με αφορμή το παραπάνω σχόλιο σχετικά με τους περιορισμούς που είχαμε λόγω hardware, θεωρείται σκόπιμο να αναφερθεί ως πρώτο κομμάτι μελλοντικής έρευνας η χρήση παράλληλων δομών hardware στην πλευρά της έμπιστης τρίτης οντότητας για την βελτίωση του μέγιστου δυνατού ρυθμού επεξεργασίας συναγερμών. Επίσης και η χρήση μεγάλων υπολογιστικών μονάδων από πλευράς των ίδιων τον οργανισμών, εκτιμάται πως θα επιφέρει σημαντικές βελτιώσεις.

Παράλληλα θα μπορούσε να δημιουργηθεί ένα σύστημα ανίχνευσης μη έντιμων συμπεριφορών των επιμέρους μελών, μέσω της χρήσης secret sharing schemes.Ενώ θα μπορούσαμε να δοκιμάσουμε την χρήση άλλων ομομορφικών κρυπταλγόριθμων με σκοπό είτε την ενίσχυση της απόδοσης είτε την περαιτέρω ενίσχυση της ιδιωτικότητας των οργανισμών. Θα μπορούσε να αναπτυχθεί ακόμα κι ένας μηχανισμός consensus μεταξύ των clients, όσον αφορά τους κανόνες ανίχνευσης εισβολών.

Τέλος αυτό που θα ολοκλήρωνε την παρούσα έρευνα δεν θα μπορούσε να είναι κάτι άλλο πέρα από την ανάπτυξη μιας ολοκληρωμένης πλατφόρμας βασισμένης στην προτεινόμενη μεθοδολογία και τους προτεινόμενους αλγορίθμους, και την εφαρμογή αυτής της πλατφόρμας σε πραγματικούς οργανισμούς ανά τον κόσμο.

ΒΙΒΛΙΟΓΡΑΦΙΑ

[1] Mauro Andreolini, Michele Colajanni, and Mirco Marchetti. A collaborative framework for intrusion detection in mobile networks. *Inf. Sci.*, 321(C):179–192, November 2015.

[2] Zvika Brakerski. Fully homomorphic encryption without modulus switching from classical gapsvp. In *Advances in cryptology–crypto 2012*, pages 868–886. Springer, 2012.

[3] Josh Benaloh Clarkson. Dense probabilistic encryption. In *In Proceedings of the Workshop on Selected Areas of Cryptography*, pages 120–128, 1994.

[4] Chandler Davis. The norm of the schur product operation. *Numerische Mathematik*, 4(1):343–344, Dec 1962.

[5] AineMac Dermott, Qi Shi, and Kashif Kifayat. Collaborative intrusion detection in federated cloud environments. *Journal of Computer Sciences and Applications*, 3(3A):10–20, 2015.

[6] Taher ElGamal. A public key cryptosystem and a signature scheme based on discrete logarithms. In George Robert Blakley and David Chaum, editors, *Advances in Cryptology*, pages 10–18, Berlin, Heidelberg, 1985. Springer Berlin Heidelberg.

[7] Craig Gentry. *A fully homomorphic encryption scheme*. PhD thesis, Stanford University, 2009. `crypto.stanford.edu/craig`.

[8] Craig Gentry. Fully homomorphic encryption using ideal lattices. In *Proceedings of the Forty-first Annual ACM Symposium on Theory of Computing*, STOC '09, pages 169–178, New York, NY, USA, 2009. ACM.

[9] Craig Gentry and Shai Halevi. Implementing gentry's fully-homomorphic encryption scheme. In *Annual International Conference on the Theory and Applications of Cryptographic Techniques*, pages 129–148. Springer, 2011.

[10] Shafi Goldwasser and Silvio Micali. Probabilistic encryption & how to play mental poker keeping secret all partial information. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 365–377, New York, NY, USA, 1982. ACM.

[11] J. Hong and C. C. Liu. Intelligent electronic devices with collaborative intrusion detection systems. *IEEE Transactions on Smart Grid*, PP(99):1–1, 2017.

[12] Roger A Horn. The hadamard product. In *Proc. Symp. Appl. Math*, volume 40, pages 87–169, 1990.

[13] Mike Ivanov. paillier. `https://github.com/mikeivanov/paillier`, 2011.

[14] Richeng Jin, Xiaofan He, and Huaiyu Dai. On the tradeoff between privacy and utility in collaborative intrusion detection systems-a game theoretical approach. In *Proceedings of the Hot Topics in Science of Security: Symposium and Bootcamp*, HoTSoS, pages 45–51, New York, NY, USA, 2017. ACM.

[15] Wenjuan Li, Weizhi Meng, Lam-For Kwok, and Horace H.S. IP. Enhancing collaborative intrusion detection networks against insider attacks usingsupervised intrusion sensitivity-based trust management model. *Journal of Network and Computer Applications*, 77:135 – 145, 2017.

[16] Hong Liang, Yufei Ge, Wenjiao Wang, and Lin Chen. Collaborative intrusion detection as a service in cloud computing environment. In *2015 IEEE International Conference on Progress in Informatics and Computing (PIC)*, pages 476–480, Dec 2015.

[17] Anderson Morais and Ana Cavalli. A distributed and collaborative intrusion detection architecture for wireless mesh networks. *Mobile Networks and Applications*, 19(1):101–120, Feb 2014.

[18] Pascal Paillier. Public-key cryptosystems based on composite degree residuosity classes. In Jacques Stern, editor, *Advances in Cryptology — EUROCRYPT '99*, pages 223–238, Berlin, Heidelberg, 1999. Springer Berlin Heidelberg.

[19] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Commun. ACM*, 21(2):120–126, February 1978.

[20] Ali Shiravi, Hadi Shiravi, Mahbod Tavallaee, and Ali A. Ghorbani. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, 31(3):357 – 374, 2012.

[21] Shalini S Singh and NC Chauhan. K-means v/s k-medoids: A comparative study. In *National Conference on Recent Trends in Engineering & Technology*, volume 13, 2011.

[22] Damien Stehlé and Ron Steinfeld. Faster fully homomorphic encryption. In *International Conference on the Theory and Application of Cryptology and Information Security*, pages 377–394. Springer, 2010.

[23] Z. Tan, U. T. Nagar, X. He, P. Nanda, R. P. Liu, S. Wang, and J. Hu. Enhancing big data security with collaborative intrusion detection. *IEEE Cloud Computing*, 1(3):27–33, Sept 2014.

[24] E. Vasilomanolakis, M. Krügl, C. G. Cordero, M. Mühlhäuser, and M. Fischer. Skipmon: A locality-aware collaborative intrusion detection system. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8, Dec 2015.

[25] Chenfeng Vincent Zhou, Christopher Leckie, and Shanika Karunasekera. Decentralized multi-dimensional alert correlation for collaborative intrusion detection. *Journal of Network and Computer Applications*, 32(5):1106 – 1123, 2009. Next Generation Content Networks.